(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－278838
（P2002－278838A）
(43)公開日　平成14年９月27日(2002.9.27)

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)
G06F	12/14	310	G06F	12/14	310K　5B017
		320			320A　5B035
	12/00	537		12/00	537A　5B058
	15/00	330		15/00	330A　5B082
G06K	17/00		G06K	17/00	S　5B085

審査請求　未請求　請求項の数46　ＯＬ　（全165頁）　最終頁に続く

(21)出願番号　特願2001－73352(P2001－73352)

(22)出願日　平成13年３月15日(2001.3.15)

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　吉野　賢治
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　石橋　義人
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　100101801
弁理士　山田　英治　（外２名）

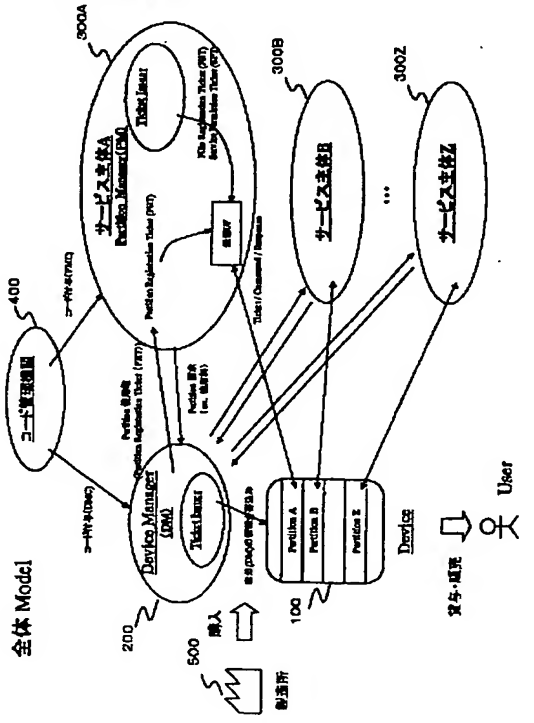
最終頁に続く

(54)【発明の名称】　メモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体

(57)【要約】

【課題】　デバイス内に生成した分割メモリ領域であるパーティションの独立した管理構成を可能としたメモリアクセス制御システムを提供する。

【解決手段】　複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティションマネージャの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する。メモリ部はパーティションマネージャによって管理されるメモリ領域としてのパーティション領域、デバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれか指定方式に従って実行することが可能とした。



【特許請求の範囲】

【請求項１】データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御システムであり前記メモリ搭載デバイスのメモリ部は、

前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、

前記メモリ搭載デバイスは、

前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成を有することを特徴とするメモリアクセス制御システム。

【請求項２】前記アクセス制御チケットは、

前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、

前記メモリ搭載デバイスは、

該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項３】前記アクセス制御チケットは、

前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、

前記メモリ搭載デバイスは、

該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項４】前記アクセス制御チケットは、

該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、

前記メモリ搭載デバイスは、

アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項５】前記アクセス制御チケットは、

該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、

前記メモリ搭載デバイスは、

アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項６】前記デバイスマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（ＰＲＴ）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からパーティション登録チケット（ＰＲＴ）を受領した場合は、

受領パーティション登録チケット（ＰＲＴ）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項７】前記パーティション登録チケット（ＰＲＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項６に記載のメモリアクセス制御システム。

【請求項８】前記パーティションマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（ＦＲＴ）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からファイル登録チケット（ＦＲＴ）を受領した場合は、

受領ファイル登録チケット（ＦＲＴ）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする請求項１に記載のメモリアクセス制御システム。

【請求項９】前記ファイル登録チケット（ＦＲＴ）は、

前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項８に記載のメモリアクセス制御システム。

【請求項１０】前記パーティションマネージャの管理するアクセス制御チケットには、

前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）を含み、

前記メモリ搭載デバイスは、

前記アクセス機器からサービス許可チケット（ＳＰＴ）を受領した場合は、
受領サービス許可チケット（ＳＰＴ）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする請求項１に記載のメモリアクセス制御システム。
【請求項１１】前記サービス許可チケット（ＳＰＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項１０に記載のメモリアクセス制御システム。
【請求項１２】前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、
前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（ＤＵＴ）を含み、
前記メモリ搭載デバイスは、
前記アクセス機器からデータアップデートチケット（ＤＵＴ）を受領した場合は、
受領データアップデートチケット（ＤＵＴ）の記録に従ったデータ更新処理を実行することを特徴とする請求項１に記載のメモリアクセス制御システム。
【請求項１３】前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、
前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項１２に記載のメモリアクセス制御システム。
【請求項１４】データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての１以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのデバイス管理を実行するデバイス管理装置であり、
前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するメモリアクセス制御チケットとしてのパーティション登録チケット（ＰＲＴ）発行手段を有することを特徴とするデバイス管理装置。
【請求項１５】前記デバイス管理装置は、
前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする請求項１４に記載のデバイス管理装置。
【請求項１６】前記パーティション登録チケット（ＰＲＴ）は、
前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする請求項１４に記載のデバイス管理装置。
【請求項１７】前記パーティション登録チケット（ＰＲＴ）は、
前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする請求項１４に記載のデバイス管理装置。
【請求項１８】前記パーティション登録チケット（ＰＲＴ）は、
該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする請求項１４に記載のデバイス管理装置。
【請求項１９】前記パーティション登録チケット（ＰＲＴ）は、
該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする請求項１４に記載のデバイス管理装置。
【請求項２０】データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての１以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのパーティション管理を実行するパーティション管理装置であり、
前記メモリ搭載デバイスのメモリ部に対して生成されたパーティション内に対するアクセスを許容するアクセス制御チケット発行手段を有することを特徴とするパーティション管理装置。
【請求項２１】前記アクセス制御チケットは、
前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（ＦＲＴ）であることを特徴とする請求項２０に記載のパーティション管理装置。
【請求項２２】前記アクセス制御チケットは、
前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）であることを特徴とする請求項２０に記載のパーティション管理装置。
【請求項２３】前記パーティション管理装置は、
前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする請求項２０に記載のパーティション管理装置。
【請求項２４】前記アクセス制御チケットは、

前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする請求項２０に記載のパーティション管理装置。

【請求項２５】前記アクセス制御チケットは、
前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする請求項２０に記載のパーティション管理装置。

【請求項２６】前記アクセス制御チケットは、
該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする請求項２０に記載のパーティション管理装置。

【請求項２７】前記アクセス制御チケットは、
該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする請求項２０に記載のパーティション管理装置。

【請求項２８】データ格納可能なメモリ部を有するメモリ搭載デバイスであり、
前記メモリ搭載デバイスのメモリ部は、
パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、
前記メモリ搭載デバイスは、
前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する制御手段を有することを特徴とするメモリ搭載デバイス。

【請求項２９】前記アクセス制御チケットは、
前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、
前記制御手段は、
該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項２８に記載のメモリ搭載デバイス。

【請求項３０】前記アクセス制御チケットは、
前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、
前記制御手段は、
該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項２８に記載のメモリ搭載デバイス。

【請求項３１】前記アクセス制御チケットは、
該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、
前記制御手段は、
アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項２８に記載のメモリ搭載デバイス。

【請求項３２】前記アクセス制御チケットは、
該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、
前記制御手段は、
アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする請求項２８に記載のメモリ搭載デバイス。

【請求項３３】データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御方法であり前記メモリ搭載デバイスのメモリ部は、
前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、
前記メモリ搭載デバイスは、
前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行することを特徴とするメモリアクセス制御方法。

【請求項３４】前記アクセス制御チケットは、
前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、
前記メモリ搭載デバイスは、
該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。

【請求項３５】前記アクセス制御チケットは、
前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、

前記メモリ搭載デバイスは、
該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項３６】前記アクセス制御チケットは、
該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、
前記メモリ搭載デバイスは、
アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項３７】前記アクセス制御チケットは、
該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、
前記メモリ搭載デバイスは、
アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項３８】前記デバイスマネージャの管理するアクセス制御チケットには、
前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（ＰＲＴ）を含み、
前記メモリ搭載デバイスは、
前記アクセス機器からパーティション登録チケット（ＰＲＴ）を受領した場合は、
受領パーティション登録チケット（ＰＲＴ）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項３９】前記パーティション登録チケット（ＰＲＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項３８に記載のメモリアクセス制御方法。
【請求項４０】前記パーティションマネージャの管理するアクセス制御チケットには、
前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（ＦＲＴ）を含み、
前記メモリ搭載デバイスは、
前記アクセス機器からファイル登録チケット（ＦＲＴ）を受領した場合は、
受領ファイル登録チケット（ＦＲＴ）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項４１】前記ファイル登録チケット（ＦＲＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする請求項４０に記載のメモリアクセス制御方法。
【請求項４２】前記パーティションマネージャの管理するアクセス制御チケットには、
前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）を含み、
前記メモリ搭載デバイスは、
前記アクセス機器からサービス許可チケット（ＳＰＴ）を受領した場合は、
受領サービス許可チケット（ＳＰＴ）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項４３】前記サービス許可チケット（ＳＰＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項４２に記載のメモリアクセス制御方法。
【請求項４４】前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、
前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（ＤＵＴ）を含み、
前記メモリ搭載デバイスは、
前記アクセス機器からデータアップデートチケット（ＤＵＴ）を受領した場合は、
受領データアップデートチケット（ＤＵＴ）の記録に従ったデータ更新処理を実行することを特徴とする請求項３３に記載のメモリアクセス制御方法。
【請求項４５】前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、
前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット

（ＤＵＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする請求項３３に記載のメモリアクセス制御方法。

【請求項４６】データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有するメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、

前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領するステップと、

アクセス機器との相互認証を実行するステップと、

受領チケットの記述に応じたチケット検証処理を実行するステップと、

受領チケットの記述に応じた処理を実行するステップと、

を有することを特徴とするプログラム記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、メモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体に関する。さらに、詳細には、１つのメモリを複数の領域（パーティション）に区分けし、各パーティション内にサービス提供者あるいは関係エンティティの管理するデータを格納して、ユーザが１つのメモリ搭載デバイスを用いて様々なサービスに供用することを可能としたメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体に関する。

【０００２】

【従来の技術】従来、メモリを保有するデバイスとしては、テープメディア、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、半導体メディア等が利用されてきた。このうち、デバイス内のメモリをセキュアに管理できるものとして半導体メディアが注目されてきている。その理由は、半導体メモリは外部から容易にアクセスさせない構造、すなわち耐タンパ構造を実現しやすいからである。

【０００３】耐タンパ構造は、例えばデバイスを半導体によるシングルチップ構成とし、該チップに制御部、メモリコントローラ、不揮発性メモリ、電圧検出手段、周波数検出手段等を備え、不揮発性メモリを外部から容易に読み書きができないようにアルミ層のようなダミー層に挟まれた構成とすることによって実現される。

【０００４】このようなセキュアデバイスの従来のメモリ構造について図９６「従来のメモリ構造」を用いて説明する。図９６のメモリは、例えば電子マネーとして利用可能なメモリ構成を示している。図９６に示すように、メモリ領域は大きく３つに別れている。すなわち、データ領域、メモリ管理領域、システム領域である。

【０００５】データ領域には各データ内の先頭に格納された「データ構造」に基づくデータが格納されており、この例では、利用者名前、住所、電話番号、金額、メモ、ログの各データが格納される。メモリ管理領域には、データ領域の各データにアクセスするための格納アドレス、アクセス方法、アクセス認証鍵等が格納されている。例えば、データ領域のデータ１（利用者名前）のアクセスは読み出し（Ｒｅａｄ）のみが、アクセス認証鍵（0123……）の利用によって可能となることが示されている。また、システム領域には、デバイス識別子（ＩＤ）、データ領域にメモリ領域を確保するための認証鍵であるメモリ管理鍵等が格納される。

【０００６】図９６に示すメモリデバイスのデータ領域は複数に分割可能であり、これらの分割データ領域を、異なるサービス主体、例えば電子マネーであればそれぞれ別の電子マネーサービス提供主体（ｅｘ．異なる銀行）が管理する構成とすることができる。各分割領域のデータは、個々のサービス提供主体の他、利用者、例えば電子マネーを利用した商品販売を行なう店舗に備えられたデバイスアクセス機器としてのリーダライタ（専用リーダライタまたはＰＣなど）によりデータの読み出し、書き込み、（ｅｘ．残金データの更新）が実行される。

【０００７】図９６に示すような複数の分割されたデータ領域を持つセキュアデバイスの管理者と利用者の関係を図９７「メモリ管理者・利用者」に示す。図９７に示すように、セキュアデバイスの発行主体であるメモリ管理者と、このメモリ管理者からメモリ領域を割り当ててもらい、その割り当てられたメモリを利用するメモリ利用者がいる。メモリ利用者としてデータ１利用者～データ６利用者がいる。メモリ利用者とは例えば前述の電子マネーの例によれば、銀行または店舗等である。

【０００８】メモリ管理者は、メモリ領域を確保するためのアクセスコントロール用のメモリ管理鍵を知っており、このメモリ管理鍵を利用して、それぞれのメモリ利用者のメモリ（分割データ領域）を割り当てる。また、メモリ利用者は各データ領域のデータにアクセスするためのアクセス認証鍵を知っており、このアクセス認証鍵を利用して、それぞれ割り当てられたデータ領域内のメモリにアクセスすることができる。アクセスの態様とし

てはデータの読み出し（Ｒｅａｄ）、書き込み（Ｗｒｉｔｅ）、残金の減額（Ｄｅｃｒｅｍｅｎｔ）など、様々であり、それぞれの処理態様に応じてアクセス認証鍵を個別に設定して個別の処理の可否を設定することができる。

【０００９】例えば図９６に示すメモリ中のデータ４は、金額データであり、図９７に示すようにデータ４の利用者はデータ４に対して減額（Ｄｅｃｒｅｍｅｎｔ）の処理と、読書き（Ｒｅａｄ／Ｗｒｉｔｅ）の処理が可能である。図９６の右下の表に示すように、データ４の減額（Ｄｅｃｒｅｍｅｎｔ）の処理と、読書き（Ｒｅａｄ／Ｗｒｉｔｅ）の処理では、アクセスキーが異なり、各処理に対応したアクセスキーを使用してメモリにアクセスすることが必要となる。

【００１０】図９８に、メモリ管理者がメモリ利用者に対してメモリデバイス内のあるデータ領域を割り当てるメモリ確保処理を説明する図を示す。図９８の「メモリの確保の方式」に示すように、メモリ管理者は、図の左側に示すメモリ確保用リーダ／ライタ（Ｒ／Ｗ：Reader／Writer）を用いて図の右側に示すメモリデバイスに対するデータ領域の確保処理を実行する。メモリ確保用リーダ／ライタ（Ｒ／Ｗ：Reader／Writer）には、メモリ管理鍵を保持するためのセキュアなＮＶＲＡＭ（Non-Volatile RAM）が備えられている。なお、メモリ確保用Ｒ／Ｗとしては、セキュアデバイスの専用の読み書きＲ／Ｗであっても、またセキュアデバイスがＵＳＢ、ＰＣＭＣＩＡなどのＩ／Ｆを持つデバイスである場合、これらのインタフェースを介して読み書き可能な装置例えばＰＣであってもよい。

【００１１】Ｒ／Ｗを用いてメモリを確保するためには、まずセキュアデバイスからデバイスＩＤを読み出す。次にＲ／Ｗ内において、メモリ管理鍵とデバイスＩＤを用いて認証鍵を生成し、生成した認証鍵を用いてセキュアデバイスと相互認証を実行する。相互認証処理は例えば共通鍵方式による相互認証（ex. ISO/IEC9798-2）に従って実行される。

【００１２】相互認証に成功した後、Ｒ／Ｗはデータ構造、データサイズ、アクセス方法、アクセス認証鍵をセッション鍵で暗号化し、必要に応じてデータ検証用のＭＡＣ（Message Authentication Code）値を付加してセキュアデバイスにコマンドを送る。コマンドを受信したセキュアデバイスは、受信データを復号し、必要に応じてデータ改竄性の検証をＭＡＣ検証処理によって実行し、その後、受信データ内のデータサイズに応じてメモリのデータ領域にメモリ領域を確保し、確保した領域にデータ構造を書き込むとともに、メモリ管理領域に確保したメモリのアドレス、アクセス方法、アクセス認証鍵を書き込む。

【００１３】このようにして、メモリデバイスには複数の分割データ領域が設定される。次に、図９９の「メモリアクセス方法」に従って、複数の分割データ領域を持つメモリデバイスに対するメモリアクセス方法について説明する。図９９の左側のリーダライタは、メモリ利用者の有するメモリアクセス用リーダライタ（Ｒ／Ｗ）であり、上述のメモリ確保用Ｒ／Ｗと同様、専用Ｒ／ＷあるいはＰＣなどで構成される。メモリアクセス用リーダライタ（Ｒ／Ｗ）には、アクセス認証鍵を保持するためのセキュアなＮＶＲＡＭが備えられている。Ｒ／Ｗを用いてセキュアデバイスのデータ領域にアクセスするためには、まずセキュアデバイスからデバイスＩＤを読み出す。次にＲ／Ｗ内において、アクセス認証鍵とデバイスＩＤを用いて認証鍵を生成し、生成した認証鍵を用いてセキュアデバイスと相互認証を実行する。相互認証に成功した後、Ｒ／Ｗはアクセス認証鍵に対応するデータ領域のデータに所定のアクセスを行なう。

【００１４】このときメモリ管理領域にはアクセス方法が規定されているため、例えば、図９９の「メモリアクセス方法」に示すように、データ４（金額データ）の減額（Decrement）用のアクセス認証に成功した場合は、データ４のデータの減額は可能であっても、加算、あるいは自由な書き換え処理はできない。このように認証処理に用いるアクセス認証鍵をそれぞれのアクセス態様に応じて異なる設定とすることにより各データの安全性を高めることができる。例えば減額処理用Ｒ／Ｗが盗難にあい、盗難にあった減額処理用Ｒ／Ｗ内のＮＶＲＡＭが見破られた場合であっても、図９９のセキュアデバイス内のデータ４（金額データ）の不正な増加処理が行われる可能性を低減することができる。

【００１５】一般に入金端末はＡＴＭと同様、セキュリティを高めることができるが、出金端末は店舗等で商品引き渡しの際の代金回収機として利用されることが多く、設置場所も様々であり、端末の盗難のリスクも高くセキュリティの度合いを高めることが困難である。従って、データアクセスに対してアクセス認証鍵を異ならせる構成が有効となる。

【００１６】

【発明が解決しようとする課題】上述した従来の分割データ領域を持つメモリデバイスの利用形態において、メモリのデータ領域の確保処理、各データ領域のアクセス処理において、それぞれ、メモリ管理鍵を用いた認証処理、あるいはアクセス認証鍵を用いた認証処理を実行することによりそれぞれの処理を実行する構成としているが、これらは具体的には、例えばＤＥＳ暗号アルゴリズムによる共通鍵を適用する構成であり、公開鍵方式による認証、あるいは公開鍵方式による検証を想定したものとはなっていない。

【００１７】上述のようにメモリ管理鍵、アクセス認証鍵に共通鍵を適用した構成では認証およびアクセス許諾が一処理で実行されるという利点はあるが、認証鍵の漏洩により、漏洩鍵によるメモリアクセスが可能となって

しまうという欠点があり、セキュリティ上問題となる。

【００１８】また、メモリデバイスに対するアクセスを実行するリーダライタ（Ｒ／Ｗ）の低コスト化を実現するために、リーダライタ（Ｒ／Ｗ）に暗号アルゴリズムを実装しない構成も想定されるが、このような構成とすれば、デバイス間との認証、通信データの暗号化の一切の処理が実行できず、ユーザの金額データ、その他ユーザのプライベート情報などを保持するデバイスに対するリーダライタとしては不適である。

【００１９】本発明は、上述のような、従来技術の現状に鑑みてなされたものであり、複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティション管理エンティテイの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する構成とすることにより、各パーティション内データの独立した管理構成を実現することを目的とする。

【００２０】また、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれか指定方式に従って実行することを可能とし、様々な環境下においてセキュアなデータ通信を実行可能としたメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体を提供することを目的とする。

【００２１】

【課題を解決するための手段】本発明の第１の側面は、データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御システムであり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成を有することを特徴とするメモリアクセス制御システムにある。

【００２２】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００２３】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００２４】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００２５】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００２６】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（ＰＲＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からパーティション登録チケット（ＰＲＴ）を受領した場合は、受領パーティション登録チケット（ＰＲＴ）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする。

【００２７】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【００２８】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ

搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（ＦＲＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からファイル登録チケット（ＦＲＴ）を受領した場合は、受領ファイル登録チケット（ＦＲＴ）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする。

【００２９】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記ファイル登録チケット（ＦＲＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【００３０】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からサービス許可チケット（ＳＰＴ）を受領した場合は、受領サービス許可チケット（ＳＰＴ）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする。

【００３１】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記サービス許可チケット（ＳＰＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【００３２】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（ＤＵＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からデータアップデートチケット（ＤＵＴ）を受領した場合は、受領データアップデートチケット（ＤＵＴ）の記録に従ったデータ更新処理を実行することを特徴とする。

【００３３】さらに、本発明のメモリアクセス制御システムの一実施態様において、前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【００３４】さらに、本発明の第２の側面は、データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての１以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのデバイス管理を実行するデバイス管理装置であり、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するメモリアクセス制御チケットとしてのパーティション登録チケット（ＰＲＴ）発行手段を有することを特徴とするデバイス管理装置にある。

【００３５】さらに、本発明のデバイス管理装置の一実施態様において、前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする。

【００３６】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする。

【００３７】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする。

【００３８】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする。

【００３９】さらに、本発明のデバイス管理装置の一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする。

【００４０】さらに、本発明の第３の側面は、データファイルを格納し、パーティション管理装置によって管理されるメモリ領域としての１以上のパーティション領域と、デバイス管理装置によって管理されるデバイスマネージャ管理領域とを有するメモリ搭載デバイスのパーティション管理を実行するパーティション管理装置であり、前記メモリ搭載デバイスのメモリ部に対して生成されたパーティション内に対するアクセスを許容するアクセス制御チケット発行手段を有することを特徴とするパーティション管理装置にある。

【００４１】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または

削除処理を許容するファイル登録チケット（ＦＲＴ）であることを特徴とする。

【００４２】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）であることを特徴とする。

【００４３】さらに、本発明のパーティション管理装置の一実施態様において、前記パーティション管理装置は、前記メモリ搭載デバイスに対する公開鍵証明書の発行管理を実行する登録局構成を有することを特徴とする。

【００４４】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含むことを特徴とする。

【００４５】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含むことを特徴とする。

【００４６】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含むことを特徴とする。

【００４７】さらに、本発明のパーティション管理装置の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含むことを特徴とする。

【００４８】さらに、本発明の第４の側面は、データ格納可能なメモリ部を有するメモリ搭載デバイスであり、前記メモリ搭載デバイスのメモリ部は、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する制御手段を有することを特徴とするメモリ搭載デバイスにある。

【００４９】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記制御手段は、該アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００５０】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記制御手段は、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００５１】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００５２】さらに、本発明のメモリ搭載デバイスの一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記制御手段は、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行する構成を有することを特徴とする。

【００５３】さらに、本発明の第５の側面は、データファイルを格納したメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御方法であり前記メモリ搭載デバイスのメモリ部は、前記データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、前記メモリ搭載デバイスは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行することを特徴とするメモリアクセス制御方法にある。

【００５４】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスとチケットを出力したアクセス機器間において実行すべき相互認証態様を指定した相互認証指定データを含み、前記メモリ搭載デバイスは、該

アクセス制御チケットの相互認証指定データに応じた相互認証を実行し、認証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【００５５】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、前記メモリ搭載デバイスの受領したアクセス制御チケットの検証態様を指定したチケット検証指定データを含み、前記メモリ搭載デバイスは、該アクセス制御チケットのチケット検証指定データに応じたチケット検証処理を実行し、検証の成立を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【００５６】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの発行手段のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの発行手段のカテゴリまたは識別子に基づいて、チケットが正当な発行手段により発行されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【００５７】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記アクセス制御チケットは、該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子を含み、前記メモリ搭載デバイスは、アクセス機器から受領したアクセス制御チケットに記述された該アクセス制御チケットの利用手段であるアクセス機器のカテゴリまたは識別子に基づいて、チケットが正当な利用手段により提供されたチケットであることの確認処理を実行し、該確認を条件として受領チケットの記録に応じた処理を実行することを特徴とする。

【００５８】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部に対するパーティションの生成処理または削除処理を許容するパーティション登録チケット（ＰＲＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からパーティション登録チケット（ＰＲＴ）を受領した場合は、受領パーティション登録チケット（ＰＲＴ）の記録に従ったパーティションの生成処理または削除処理を実行することを特徴とする。

【００５９】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティション登録チケット（ＰＲＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする。

【００６０】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内に生成されたパーティション内に対するデータファイルの生成処理または削除処理を許容するファイル登録チケット（ＦＲＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からファイル登録チケット（ＦＲＴ）を受領した場合は、受領ファイル登録チケット（ＦＲＴ）の記録に従ったファイルの生成処理または削除処理を実行することを特徴とする。

【００６１】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記ファイル登録チケット（ＦＲＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行される構成であることを特徴とする。

【００６２】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内のパーティション内のデータファイルに対するアクセスを許容するサービス許可チケット（ＳＰＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からサービス許可チケット（ＳＰＴ）を受領した場合は、受領サービス許可チケット（ＳＰＴ）の記録に従ったデータファイルに対するアクセス処理を実行することを特徴とする。

【００６３】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記サービス許可チケット（ＳＰＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとしてのアクセス機器に対して発行されることを特徴とする。

【００６４】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャまたは前記パーティションマネージャの管理するアクセス制御チケットには、前記メモリ搭載デバイスのメモリ部内の格納データの更新処理を許容するデータアップデートチケット（ＤＵＴ）を含み、前記メモリ搭載デバイスは、前記アクセス機器からデータアップデートチケット（ＤＵＴ）を受領した場合は、受領データアップデートチケット（ＤＵＴ）の記録に従ったデータ更新処理を実行することを特徴とする。

【００６５】さらに、本発明のメモリアクセス制御方法の一実施態様において、前記デバイスマネージャの管理するデバイスマネージャ管理領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記デバイスマネージャの管理するチケット発行手段から前記デバイスマネージャの管理するチケットユーザとしてのアクセス機器に対して発行され、前記パーティションマネージャの管理するパーティション領域のデータ更新用のデータアップデートチケット（ＤＵＴ）は、前記パーティションマネージャの管理するチケット発行手段から前記パーティションマネージャの管理するチケットユーザとして

のアクセス機器に対して発行されることを特徴とする。

【００６６】さらに、本発明の第６の側面は、データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有するメモリ部を有するメモリ搭載デバイスに対するメモリアクセス制御処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、前記メモリ部に対するアクセス制御チケットとして、前記デバイスマネージャの管理するアクセス制御チケット、または前記パーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領するステップと、アクセス機器との相互認証を実行するステップと、受領チケットの記述に応じたチケット検証処理を実行するステップと、受領チケットの記述に応じた処理を実行するステップと、を有することを特徴とするプログラム記憶媒体にある。

【００６７】なお、本発明のプログラム記憶媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。媒体は、ＣＤやＦＤ、ＭＯなどの記録媒体、通信可能媒体など、その形態は特に限定されない。

【００６８】このようなプログラム記憶媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムの機能を実現するための、コンピュータ・プログラムと記憶媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、該記憶媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【００６９】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【００７０】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態について詳細に説明する。なお、説明は、以下の項目に従って行なう。

Ａ．デバイスを利用したデータ処理システムの構成エンティティおよびチケットに関する説明

Ａ１．メモリ搭載デバイスを利用したデータ管理システムの概要

Ａ２．デバイスの構成

Ａ３．デバイスマネージャの構成

Ａ４．パーティションマネージャの構成

Ａ５．チケットユーザ（デバイスアクセス機器としてのリーダライタ）の構成

Ａ６．公開鍵証明書

Ａ７．デバイスのメモリ部における格納データ

Ａ７．１．デバイス固有情報およびデバイス内パーティション情報領域

Ａ７．２．パーティション領域

Ａ８．各チケットのデータフォーマット

Ａ８．１．パーティション登録チケット（ＰＲＴ）

Ａ８．２．ファイル登録チケット（ＦＲＴ）

Ａ８．３．サービス許可チケット（ＳＰＴ）

Ａ８．４．データアップデートチケット（ＤＵＴ）

Ｂ．ユーザに対するデバイスの配布、デバイスに対する各種設定、デバイス利用処理の詳細についての説明

Ｂ１．デバイス初期登録から利用までの流れ

Ｂ２．デバイス製造エンティティによる初期登録処理

Ｂ３．デバイスマネージャの管轄処理

Ｂ３．１．デバイスマネージャによるデバイス登録処理

Ｂ３．２．デバイスマネージャ管理下における公開鍵証明書発行処理

Ｂ４．パーティションマネージャの管轄処理

Ｂ４．１．パーティションマネージャ管理下におけるパーティション登録チケット（ＰＲＴ）を利用したパーティション設定登録、削除処理

Ｂ４．２．パーティションマネージャ管理下における公開鍵証明書発行処理

Ｂ４．３．パーティション生成処理各方式における処理手順

Ｂ４．４．ファイル登録チケット（ＦＲＴ）を利用したファイル生成、消去処理

Ｂ４．５．ファイル生成処理各方式における処理手順

Ｂ４．６．サービス許可チケット（ＳＰＴ）を利用したサービス（ファイルアクセス）処理

Ｂ４．７．サービス許可チケット（ＳＰＴ）を利用したアクセス処理各方式における処理手順

Ｂ５．データアップデートチケット（ＤＵＴ）を利用したデバイスのデータ更新処理

【００７１】

【実施例】［Ａ１．メモリ搭載デバイスを利用したデータ管理システムの概要］図１に本発明のデータ管理システムの概要を説明する図を示す。メモリ搭載デバイス（以下デバイス）１００はデバイス製造エンティティ（manufacturer）５００により製造され、デバイス管理エンティティとしてのデバイスマネージャ（ＤＭ：Device Manager）２００の管理の下にユーザに提供されて利用される。ユーザに対するデバイスの提供形態は、貸与または販売（譲渡を含む）など、いずれの形態でもよい。

【００７２】デバイス１００は、メモリ領域が複数のデ

ータ格納領域としてのパーティションに分割され、個々のパーティション(Partition A,B…Z)は、様々なサービス主体（A，B，…Z）３００Ａ～３００Ｚとしてのパーティションマネージャの管理の下、様々なサービスに利用される。

【００７３】デバイス１００に対するパーティションの設定登録処理、デバイスに設定されたパーティション内におけるファイルの設定登録処理、さらに、登録された各ファイルに対するアクセス処理にはそれぞれ正当なチケット発行手段（Ticket Issuer）の発行したデバイスに対するアクセスコントロールチケットを必要とする。

【００７４】デバイス１００に対するパーティションの設定登録処理には、正当なチケット発行手段（Ticket Issuer）の発行したパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）が必要であり、デバイスに設定されたパーティション内に対するファイルの設定登録処理には、正当なチケット発行手段（Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ：File Registration Ticket）が必要であり、また、各ファイルに対するアクセスには正当なチケット発行手段（Ticket Issuer）の発行したサービス許可チケット（ＳＰＴ：Service Permission Ticket）が必要となる。

【００７５】各チケットには、デバイス１００に対するアクセスルール、例えばデバイスに対して読み書きなど各種処理を実行するリーダ／ライタとデバイス間の相互認証処理に関するルールの他、例えばパーティション登録チケット（ＰＲＴ）であれば、設定できるパーティションサイズ、ファイル登録チケット（ＦＲＴ）であれば、設定できるファイルサイズ、サービス許可チケット（ＳＰＴ）であれば、実行可能なアクセス態様（ｅｘ．データ読み出し、書き込みなど）などが格納され、さらにチケット発行者、チケット利用者に関する情報、その他の情報が格納される。また、これらチケット格納データに対する改竄チェック用のＩＣＶ（Integrity Check Value）が記録され、チケットの改竄の無いことを条件としてチケットに記録された範囲内の処理が実行可能となる。これらチケットの詳細については後段で説明する。

【００７６】図１において示す例では、パーティション登録チケット（ＰＲＴ）を発行するチケット発行手段（Ticket Issuer）はデバイスマネージャ（DM）２００内に設定され、パーティションマネージャとしてのサービス主体Ａ，３００Ａ内にファイル登録チケット（ＦＲＴ）、およびサービス許可チケット（ＳＰＴ）を発行するチケット発行手段（Ticket Issuer）が設定される。なお図１の構成は、サービス主体Ｂ…Ｚ，３００Ｂ～３００Ｚについても基本的にサービス主体Ａと同様の構成を持つものであり、各サービス主体にファイル登録チケット（ＦＲＴ）、およびサービス許可チケット（ＳＰＴ）を発行するチケット発行手段（TicketIssuer）が設定される。

【００７７】なお、図１では、サービス主体とパーティションマネージャ（ＰＭ）を同一エンティテイとして示しているが、必ずしもこれらのエンティテイは同一であることは必要でなく、デバイスに設定されるメモリ領域としてのパーティションを管理するパーティションマネージャと、パーティションマネージャの管理するメモリ領域であるパーティションをパーティションマネージャから所定契約の下に借り受けて、借り受けたパーティション内に様々なファイルを格納してサービスを提供するサービス主体とが別エンティテイとして存在してもよい。以下の説明では、説明を簡略化するためにサービス主体がパーティションマネージャとして機能する構成例について説明する。

【００７８】各サービス主体３００Ａ～３００Ｚとしてのパーティションマネージャ（ＰＭ）は、デバイスマネージャ（DM）２００に対して、例えば相応の対価を支払うなど所定の契約の下に、パーティション登録チケット（ＰＲＴ）の発行要求を行ない、デバイスマネージャ（DM）の許諾の下、デバイスマネージャ（DM）内のチケット発行手段（Ticket Issuer）が各サービス主体としてのパーティションマネージャ（ＰＭ）に対してパーティション登録チケット（ＰＲＴ）を発行する。

【００７９】各サービス主体（パーティションマネージャ（ＰＭ））３００は、通信インタフェース（Ｉ／Ｆ）を介してユーザの所有デバイス１００に対するアクセスを実行し、デバイスマネージャ（DM）２００から受領したパーティション登録チケット（ＰＲＴ）に記録されたルールに従った認証、検証等の処理を実行し、かつパーティション登録チケット（ＰＲＴ）に記録された許可範囲内のパーティションの設定登録処理を実行する。この処理については後段で詳細に説明する。

【００８０】通信Ｉ／Ｆは、有線、無線を問わず、外部機器（デバイス）とのデータ通信可能なインタフェースであればよく、例えば、デバイスがＵＳＢ接続構成を持つ場合はＵＳＢＩ／Ｆ、また、ＩＣカード型であれはＩＣカード用リーダライタ、さらに公衆回線、通信回線、インターネットなど各種の通信機能を持つデバイス、あるいはこれらの通信装置に接続可能なデバイスであれば、各通信方式に従ったデータ通信Ｉ／Ｆとして構成される。

【００８１】また、デバイス１００にサービス主体３００のパーティションが設定されると、各サービス主体３００は、通信インタフェース（Ｉ／Ｆ）を介してユーザ所有のデバイス１００にアクセスし、各サービス主体３００のチケット発行手段（Ticket Issuer）の発行するファイル登録チケット（ＦＲＴ）に記録されたルールに従った認証、検証等の処理を実行し、かつファイル登録チケット（ＦＲＴ）に記録された許可範囲内のファイル

の設定登録処理を実行する。この処理については後段で詳細に説明する。

【００８２】さらに、各サービス主体３００は、通信インタフェース（Ｉ／Ｆ）を介してユーザの所有デバイス１００にアクセスし、各サービス主体のチケット発行手段（Ticket Issuer）の発行するサービス許可チケット（ＳＰＴ）に記録されたルールに従った認証、検証等の処理を実行し、かつサービス許可チケット（ＳＰＴ）に記録された許可範囲内のアクセス（ｅｘ．データの読み取り、書き込みなど）処理を実行する。この処理については後段で詳細に説明する。

【００８３】また、図１に示すように、デバイスマネージャ２００、パーティションマネージャ３００の上位にコード管理機関４００が設定され、個々のデバイスマネージャ、パーティションマネージャに各エンティテイの識別情報としてのコードを割り振る処理を行なっている。これら各マネージャに付与されたコードは、前述のパーティション登録チケット（ＰＲＴ）、ファイル登録チケット（ＦＲＴ）等のアクセスコントロールチケットの格納データとされる。

【００８４】デバイス１００がユーザに提供（ｅｘ．貸与、販売）されユーザが利用する以前に、提供デバイスを管理するデバイスマネージャ（ＤＭ）２００が設定され、その提供デバイス内にデバイスマネージャコード他、デバイスマネージャの管理情報が書き込まれる。これらのデータ詳細については後述する。

【００８５】本発明のメモリデバイスを利用したデータ管理システムにおける公開鍵証明書の発行処理と各エンティテイの関係について、図２を用いて説明する。

【００８６】図２は、デバイス管理エンティテイとしてのデバイスマネージャ、デバイスに設定された各パーティションの管理エンティテイとして、２つのパーティションマネージャ３００Ａ，３００Ｂ、デバイスマネージャ２００に対して識別コードを付与するコード管理機関４００を示している。さらに、デバイスマネージャ２００の管轄する登録局２１０からの公開鍵証明書発行要求に応じて、デバイスマネージャ２００、デバイスマネージャ管轄の各機器（パーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）２１０、あるいはデバイス１００に対応するデバイス対応公開鍵証明書（CERT-DEV）を発行するデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ）：Certificate Authority）６１０、パーティションマネージャ３００Ａ，３００Ｂ管轄の各機器（ファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）３１０、サービス許可チケット発行手段（ＳＰＴ）３２０、チケットユーザであるデバイスアクセス機器としてのリーダライタ７１１～７１４、あるいはデバイス１００のパーティションに対応するパーティション対応公開鍵証明書（CERT-PAR）を発行するパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ）：Certificate Authority）６２０，６３０が存在する。

【００８７】なお、図２には、認証局をデバイスマネージャ対応認証局：ＣＡ（Certificate Authority）ｆｏｒ ＤＭ（またはＣＡ（ＤＥＶ））６１０と、パーティションマネージャ対応認証局：ＣＡ ｆｏｒ ＰＡＲ（またはＣＡ（ＰＡＲ））６２０，６３０と個別に有する構成を示しているが、両機能を持つ唯一の認証局を設けたり、複数のパーティションマネージャに対応する共通の認証局とデバイスマネージャ対応認証局を別々に設けたり、その構成は自由である。

【００８８】デバイスマネージャ２００、パーティションマネージャ３００Ａ，３００Ｂは、自己の公開鍵証明書、各マネージャの管理する各機器（チケット発行手段、チケットユーザ）の公開鍵証明書、または、デバイス１００からの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局（ＲＡ：Registration Authority）２２０，３３０を有する。

【００８９】これら登録局（ＲＡ）２２０，３３０を介して各認証局（ＣＡ）６１０，６２０，６３０から発行された公開鍵証明書はデバイス１００に格納され、デバイス１００に対する処理としての例えばパーティションの設定処理、あるいはパーティションに対する処理としての例えばファイル設定処理、さらにファイルに対するアクセス処理等の際の相互認証処理、あるいは前述した各チケットの正当性検証処理に使用される。これら公開鍵証明書の発行処理、公開鍵証明書を使用した各処理の詳細については後述する。

【００９０】図２において、デバイス１００は、パーティションとしてパーティションマネージャ１，３００Ａの管理パーティション：ＰＭ１Ａｒｅａ、パーティションマネージャ２，３００Ｂの管理パーティション：ＰＭ２Ａｒｅａを有し、さらに、デバイスマネージャ２００の管理領域としてのＤＭＡｒｅａを有する。

【００９１】デバイスマネージャ２００はパーティション登録チケット発行手段（ＰＲＴ Issuer）２１０を有し、パーティションマネージャ３００は、ファイル登録チケット発行手段（ＦＲＴ Issuer）３１０、およびサービス許可チケット発行手段（ＳＰＴ Issuer）３２０を有しており、それぞれ各チケットを発行する。

【００９２】パーティションマネージャ１，３００Ａは、ＰＲＴ、ＦＲＴ、ＳＰＴ各チケット毎に異なる専用のリーダ／ライタ（デバイスに対するデータ読み出し書き込み用のインタフェース）７１１～７１３を有した構成であり、パーティションマネージャ２，３００Ｂは、各チケットに共通のリーダ／ライタ７１４を有した構成を示している。リーダ／ライタはこのように様々な構成をとることが可能である。

【００９３】さらに図３を用いてエンティテイの具体例

について説明する。図３には、デバイスに設定されたパーティションを利用したサービスを提供するサービス主体としてのパーティションマネージャとして東西鉄道株式会社および南北鉄道株式会社の２つのサービス主体を想定し、これらパーティションマネージャに対してパーティションの設定登録を行なうデバイスマネージャとして日本鉄道グループという組織を想定したデバイス利用構成例を示している。

【００９４】東西鉄道株式会社は、ユーザのデバイスに設定された自身の管理するパーティション：ＰＭ１内に複数のファイルを設定登録している。すなわち、定期券用ファイル、プリペイド用ファイル、その他用ファイルである。各サービス主体としてのパーティションマネージャは自己の提供するサービスに応じて設定されたデバイスマネージャによって割り当てられたパーティション内に様々なファイルを登録できる。ただし、ファイルの設定登録にはファイル登録チケット（ＦＲＴ）が必要となる。

【００９５】東西鉄道株式会社は、デバイスの１つのパーティション：ＰＭ１Ａｒｅａを管理するパーティションマネージャとして機能する。パーティション：ＰＭ１Ａｒｅａは、デバイスマネージャとしての日本鉄道グループによって、日本鉄道グループのＰＲＴＩｓｓｕｅｒの発行するパーティション登録チケット（ＰＲＴ）に記録されたルールに従った認証、検証等の処理が実行され、かつパーティション登録チケット（ＰＲＴ）に記録された許可範囲内のパーティションの設定登録処理によって設定されて、東西鉄道株式会社に付与される。

【００９６】東西鉄道株式会社は、付与されたパーティション：ＰＭ１Ａｒｅａに自身の提供するサービスに応じて様々なファイルを設定する。例えば定期券ファイル、プリペイド用ファイルであり、定期券ファイル内のデータ格納エリアには例えば、定期券使用者名、使用期間、利用区間など定期券管理データとして必要な各種データを記録する。また、プリペイド用ファイルには、使用者名、プリペイド金額、残額データなどが記録される。このファイル設定処理には、東西鉄道のＦＲＴＩｓｓｕｅｒの発行するファイル登録チケット（ＦＲＴ）に記録されたルールに従った認証、検証等の処理が実行され、かつファイル登録チケット（ＦＲＴ）に記録された許可範囲内のファイルの設定登録処理によって設定される。

【００９７】このように様々なファイルの設定されたデバイスがユーザによって使用される。例えば、ユーザがデバイスを使用してデバイスアクセス機器としてのリーダライタを備えた改札にデバイスをセットして利用することが可能である。例えば改札に備えられた正当なリーダライタにより、定期券用ファイルのアクセスが実行されて、利用区間の読み取りが行われる。またプリペイド用ファイルにアクセスして、プリペイド用ファイル内の残金データの更新処理が実行される。

【００９８】デバイス内の、いずれのファイルにアクセスしてどのような処理（読み取り、書き込み、、減額ｅｔｃ）を実行するかは、東西鉄道のサービス許可チケット（ＳＰＴ）発行手段（ＳＰＴ issuer）の発行するサービス許可チケット（ＳＰＴ）に記録されている。例えば改札に備えられた正当なデバイスアクセス機器としてのリーダライタにはこれらのチケットが格納されており、チケットに記録されたルールに従ってデバイス間との認証処理、チケット検証等の処理が実行される。デバイスアクセス機器としてのリーダライタおよびデバイス相互が正当な機器であり、使用チケットが正当である場合にサービス許可チケット（ＳＰＴ）に記録された許可範囲内の処理（ｅｘ．ファイル内のデータ読み取り、書き込み、、減額ｅｔｃ）が実行されることになる。

【００９９】パーティション登録チケット（ＰＲＴ）、ファイル登録チケット（ＦＲＴ）、およびサービス許可チケット（ＳＰＴ）の各種チケットを発行するチケット発行手段（Ticket Issuer）とチケット発行手段によって発行されたチケットを利用するチケットユーザ（Ticket User）の一般的な対応関係を図４に示す。

【０１００】チケット発行手段（Ticket Issuer）は、図１他で説明したように、デバイスマネージャ、あるいはパーティションマネージャの管理下にあり、デバイスに対する処理に応じたパーティション登録チケット（ＰＲＴ）、ファイル登録チケット（ＦＲＴ）、およびサービス許可チケット（ＳＰＴ）の各種チケットを発行する。チケットユーザ（Ticket User）は、チケット発行手段によって発行されたチケットを利用する機器、手段であり、具体的には例えばデバイスに対するデータ書き込み、読み取りなどの処理を実行するデバイスアクセス機器としてのリーダライタ等の機器が相当する。

【０１０１】図４に示すように、チケットユーザは、複数のチケットを格納して使用することが可能である。また、単一のチケット、例えば図３を用いて説明した定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（ＳＰＴ）のみを格納し、区間データ読み取りのみの処理を実行する構成とすることも可能である。

【０１０２】例えば、あるサービス主体（パーティションマネージャ）である鉄道会社の定期券読み取り専用の改札には、上述の定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（ＳＰＴ）のみを格納したデバイスアクセス機器としてのリーダライタを設定して、ユーザが所有するデバイスから区間データの読み取り処理を実行する。例えば、定期券、プリペイド双方の処理を実行する改札のデバイスアクセス機器としてのリーダライタには上述の定期券用ファイルの区間データ読み取りのみの実行を許可したサービス許可チケット（ＳＰＴ）、およびプリペイド用ファイル

の残金データ減額処理を許可したサービス許可チケット（ＳＰＴ）を併せて格納し、定期券用ファイル、およびプリペイド用ファイル両ファイルに対する処理を実行可能とすることも可能である。

【０１０３】また、パーティション登録チケット（ＰＲＴ）、ファイル登録チケット（ＦＲＴ）、およびサービス許可チケット（ＳＰＴ）を格納し、パーティション登録、ファイル登録、ファイル内のデータアクセス等のすべての処理を実行可能としたチケットユーザ（ｅｘ．リーダライタ）を構成することも可能である。このようにチケットユーザの実行可能な処理は、チケットユーザが適用可能なチケットによって決定されることになる。

【０１０４】［Ａ２．デバイスの構成］次に、上述の複数のパーティションにデータ格納領域を分割されたメモリを持つデバイスについて説明する。図５にデバイスの構成図を示す。

【０１０５】図５に示すように、デバイス１００は、プログラム実行機能、演算処理機能を持つＣＰＵ（Central Processing Unit）１０１を有し、デバイスアクセス機器としてのリーダライタ等、外部機器との通信処理用のインタフェース機能を持つ通信インタフェース１０２、ＣＰＵ１０１によって実行される各種プログラム、例えば暗号処理プログラムなどを記憶したＲＯＭ（Read Only Memory）１０３、実行プログラムのロード領域、また、各プログラム処理におけるワーク領域として機能するＲＡＭ（Random Access Memory）１０４、外部機器との認証処理、電子署名の生成、検証処理、格納データの暗号化、復号処理等の暗号処理を実行する暗号処理部１０５、前述したパーティション、ファイルが設定格納されるとともに、各種鍵データを含むデバイスの固有情報を格納した例えばＥＥＰＲＯＭ(Electrically Erasable Programmable ROM)によって構成されるメモリ部１０６を有する。メモリ部１０６（ｅｘ．ＥＥＰＲＯＭ）１０６に格納される情報については、後段で詳述する。

【０１０６】メモリ部１０６のデータ格納構成を図６に示す。メモリ部は例えば、ＥＥＰＲＯＭ(Electrically Erasable Programmable ROM)と呼ばれる電気的に書き換え可能な不揮発性メモリの一形態であるフラッシュメモリである。

【０１０７】図６に示すように、本実施例においては、１ブロック３２バイト、ブロック数０ｘＦＦＦＦのデータ格納領域を持ち、主要領域としてパーティション領域、未使用領域、デバイス固有情報およびデバイス内パーティション情報領域を持つ。

【０１０８】パーティション領域には、前述のパーティションマネージャによる管理領域であるパーティションが設定登録される。なお、図６に示すメモリは既にパーティションが設定された例を示しているが、新規に製造されたデバイスには、パーティションが設定されておらずパーティション領域は存在しない。パーティションは、前述したように、デバイスマネージャの管理するパーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）の発行したＰＲＴチケットに基づいて各サービス主体としてのパーティションマネージャが所定の手続き、すなわちパーティション登録チケット（ＰＲＴ）に設定されたルールに従ってデバイス内のメモリに設定する。

【０１０９】デバイス固有情報およびデバイス内パーティション情報領域には、デバイス製造エンティテイの情報、デバイスマネージャに関する情報、設定パーティション情報、デバイスに対するアクセスを実行してパーティションの設定登録処理を実行する際に必要となる鍵情報などが格納される。これら格納情報の詳細については後述する。なお、デバイス固有情報領域の格納データは、後述する相互認証時に適用するデバイスの固有値としてのＩＤｍに対応するデータとして使用可能である。

【０１１０】また、図に示すようにパーティション領域は、さらに１以上のファイル領域、未使用領域、パーティション固有情報およびパーティション内ファイル領域を有する。ファイル領域は、パーティションマネージャであるサービス主体が、例えば前述したような定期券用、プリペイド用などサービス毎に設定したファイルを格納する領域である。未使用領域は、さらにファイル設定可能な領域である。パーティション固有情報およびパーティション内ファイル情報領域は、例えばパーティション内のファイルに関する情報、ファイルアクセス処理に必要となる鍵情報などが格納される。これら格納情報の詳細については後述する。

【０１１１】［Ａ３．デバイスマネージャの構成］次に、デバイスマネージャの構成について図７を用いて説明する。デバイスマネージャは、ユーザに提供（販売または貸与）されるデバイスの管理エンティテイである。

【０１１２】デバイスマネージャ２００は、デバイス内のメモリ部の分割領域として設定されるパーティションを利用してサービスを提供するサービス主体としてのパーティションマネージャからの要求に応じてデバイスに対するパーティション設定を可能化するパーティション登録チケット（ＰＲＴ）を発行するパーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）２１０を有する。

【０１１３】さらに、デバイスマネージャ２００は、デバイスに対応するデバイス対応公開鍵証明書（CERT-DEV）を発行する。デバイスマネージャ２００は、デバイスからの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局（ＣＡ（ＤＥＶ）：Certificate Authority）６１０に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局（ＲＡ：Registration Authority）２２０としての機能を有する。

【０１１４】図７に示すように、デバイスマネージャ２

００のパーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）２１０は、制御手段２１１と、データベース２１２を有し、データベース２１２は、パーティション登録チケット（ＰＲＴ）の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のパーティションマネージャ識別子、チケット識別子、チケットユーザ（ｅｘ．リーダライタ，ＰＣ，ｅｔｃ）識別子などを対応付けたデータが格納される。

【０１１５】また、登録局（ＲＡ：Registration Authority）２２０は、制御部２２１、公開鍵証明書の発行管理用のデータベース２２２を有し、公開鍵証明書の発行管理データとして、例えば公開鍵証明書を発行したデバイス識別子、公開鍵証明書の識別子（シリアルナンバ）等を対応付けたデータが格納される。

【０１１６】デバイスマネージャ２００のパーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）２１０の制御手段２１１は、パーティションマネージャとのデータ通信により、パーティション登録チケット（ＰＲＴ）の発行処理を実行する。また、登録局（ＲＡ：Registration Authority）２２０の制御手段２２１は、デバイスに対する公開鍵証明書の発行処理を実行し、この際、デバイスとの通信、デバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））６１０との通信を実行する。これらの処理の詳細については後段で説明する。ここでは、制御手段２１１，２２１の構成について図８を用いて説明する。

【０１１７】制御手段２１１，２２１はいずれも例えばデータ処理手段としてのＰＣと同様の構成により実現され、具体的には例えば図８に示すような構成を持つ。制御手段の構成について説明する。制御部２１１１は各種処理プログラムを実行する中央演算処理装置（ＣＰＵ:Central Processing Unit）によって構成される。ＲＯＭ（Read only Memory）２１１２は、暗号処理プログラム等の実行処理プログラムを記憶したメモリである。ＲＡＭ（Random Access Memory）２１１３は、制御部２１１１が実行するプログラム、例えばデータベース管理プログラム、暗号処理プログラム、通信プログラム等、実行プログラムの格納領域、またこれら各プログラム処理におけるワークエリアとして使用される。

【０１１８】表示部２１１４は、液晶表示装置、ＣＲＴなどの表示手段を有し、制御部２１１１の制御の下、様々なプログラム実行時のデータ、例えば処理対象のデータ内容等を表示する。入力部２１１５は、キーボードや、例えばマウス等のポインティングデバイスを有し、これら各入力デバイスからのコマンド、データ入力を制御部２１１１に出力する。ＨＤＤ（Hard Disk Drive）２１１６は、データベース管理プログラム、暗号処理プログラム、通信プログラム等のプログラム、さらに各種データが格納される。

【０１１９】ドライブ２１１７は、例えばＨＤ（Hard Disk）や、ＦＤ（Floppy Disk）等の磁気ディスク、ＣＤ－ＲＯＭ（Compact Disk ROM）などの光ディスク、ミニディスク等の光磁気ディスク、ＲＯＭやフラッシュメモリなどの半導体メモリ等の各種記録媒体に対するアクセスを制御する機能を持つ。磁気ディスク等の各種記録媒体はプログラム、データ等を記憶する。通信インタフェース２１１８は、ネットワーク、ケーブル接続、電話回線等の有線、無線を介した通信のインタフェースとして機能し、ユーザのデバイス、パーティションマネージャ、認証局等の各エンティティとの通信インタフェースとして機能する。

【０１２０】［Ａ４．パーティションマネージャの構成］次に、パーティションマネージャの構成について図９を用いて説明する。パーティションマネージャは、ユーザに提供（販売または貸与）されるデバイスに設定されたパーティションの管理エンティティである。

【０１２１】パーティションマネージャ３００は、デバイスマネージャから付与されたパーテイション登録チケット（ＰＲＴ）を用いて、付与されたＰＲＴに記録されたルールに従って、ユーザのデバイス内のメモリ部に分割領域としてパーティションを設定し、設定されたパーティションを利用したサービスを提供する。

【０１２２】設定されたパーティションにはサービス、データに応じたファイルを設定することが可能である。ただし、ファイル設定処理には、ファイル登録チケット（ＦＲＴ）を取得することが必要であり、ファイル内のデータの読み出し、書き込みなどのデータアクセスにはサービス許可チケット（ＳＰＴ）を取得することが必要である。ファイル設定、データアクセス処理はチケットユーザ、すなわち具体的には、例えば専用のデバイスアクセス機器としてのリーダライタなどがチケットを使用して実行する。

【０１２３】パーティションマネージャ３００は、このようなチケットユーザに対するチケット発行処理手段としてのファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）３１０、およびサービス許可チケット（ＳＰＴ）発行手段（ＳＰＴ Issuer）３２０を有する。

【０１２４】さらに、パーティションマネージャ３００は、デバイスの各パーティションに対応するパーティション対応公開鍵証明書（CERT-PAR）を発行する。パーティションマネージャ３００は、デバイスからの公開鍵証明書発行要求を受理し、受理した発行要求の検証を行ない、検証の後、証明書発行要求を認証局（ＣＡ（ＰＡＲ）：Certificate Authority）６２０に対して転送する処理を行なうとともに、発行された公開鍵証明書の管理処理を行なう登録局（ＲＡ：Registration Authority）３３０としての機能を有する。

【０１２５】図９に示すように、パーティションマネージャ３００のファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）３１０は、制御手段３１１と、デー

タベース３１２を有し、データベース３１２は、ファイル登録チケット（ＦＲＴ）の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のチケットユーザ（ｅｘ．リーダライタ，ＰＣ，ｅｔｃ）識別子、チケット識別子などを対応付けたデータを格納する。

【０１２６】さらにパーティションマネージャ３００のサービス許可チケット（ＳＰＴ）発行手段（ＳＰＴ Issuer）３２０は、制御手段３２１と、データベース３２２を有し、データベース３２２は、サービス許可チケット（ＳＰＴ）の発行管理データとして、チケットの発行管理用のデータ、例えば、チケット発行先のチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ，ＰＣ，ｅｔｃ）識別子、チケット識別子などを対応付けたデータを格納する。

【０１２７】また、登録局（ＲＡ：Registration Authority）３３０は、公開鍵証明書の発行管理用のデータベース３３２を有し、公開鍵証明書の発行管理データとして、例えば公開鍵証明書を発行したデバイス識別子、パーティション識別子、公開鍵証明書の識別子（シリアルナンバ）等を対応付けたデータが格納される。

【０１２８】パーティションマネージャ３００のファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）３１０の制御手段３１１は、チケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ，ＰＣ，ｅｔｃ）とのデータ通信により、ファイル登録チケット（ＦＲＴ）の発行処理を実行し、サービス許可チケット（ＳＰＴ）発行手段（Ticket Issuer）３２０の制御手段３２１は、チケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ，ＰＣ，ｅｔｃ）とのデータ通信により、サービス許可チケット（ＳＰＴ）の発行処理を実行する。また、登録局（ＲＡ：Registration Authority）３３０の制御手段３３１は、デバイスに対する公開鍵証明書の発行処理を実行し、この際、デバイスとの通信、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））６２０との通信を実行する。これらの処理の詳細については後段で説明する。

【０１２９】なお、パーティションマネージャ３００の制御手段３１１，３２１，３３１の構成は、図８を用いて説明した前述のデバイスマネージャにおける制御手段と同様の構成であるので説明を省略する。

【０１３０】［Ａ５．チケットユーザ（デバイスアクセス機器としてのリーダライタ）の構成］デバイスアクセス機器としてのリーダライタはデバイスに対するパーティションの設定、ファイルの設定、データの読み取り、書き込み、金額データの減算、加算などの様々な処理を実行する機器として構成される。デバイスに対する処理は、処理の際に適用するパーテイション登録チケット（ＰＲＴ）、ファイル登録チケット（ＦＲＴ）、またはサービス許可チケット（ＳＰＴ）に記録されたルールに従う。すなわち、デバイスに対するすべての処理はこれら適用するチケットによって制限される。

【０１３１】デバイスアクセス機器としてのリーダライタの構成例を図１０に示す。図１０に示すように、リーダライタ７００は、プログラム実行機能、演算処理機能を持つＣＰＵ（Central Processing Unit）７０１を有し、デバイス、チケット発行手段（Ticket Issuer）等、外部機器との通信処理用のインタフェース機能を持つ通信インタフェース７０２、ＣＰＵ７０１によって実行される各種プログラム、例えば暗号処理プログラムなどを記憶したＲＯＭ（Read Only Memory）７０３、実行プログラムのロード領域、また、各プログラム処理におけるワーク領域として機能するＲＡＭ（Random Access Memory）７０４、外部機器との認証処理、電子署名の生成、検証処理、格納データの暗号化、復号処理等の暗号処理を実行する暗号処理部７０５、認証処理、暗号化、復号処理用の各種鍵データ、およびリーダライタの固有情報を格納した例えばＥＥＰＲＯＭ(Electrically ErasableProgrammable ROM)によって構成されるメモリ部７０６を有する。

【０１３２】［Ａ６．公開鍵証明書］本発明のパーティション分割メモリ領域を持つデバイスの利用において、各エンティテイ、チケット発行手段、チケットユーザ、デバイス等の相互間におけるデータ通信においては、データ送信側とデータ受信側とが互いに正規なデータ送受信対象であることを確認した上で、必要な情報を転送する、データ転送の際のセキュリティ構成を実現する手法として、転送データの暗号化処理、データに対する署名生成、検証処理が適用される。

【０１３３】暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【０１３４】暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類があるが、その１つの例としていわゆる公開鍵暗号方式と呼ばれる方式がある。公開鍵暗号方式は、発信者と受信者の鍵を異なるものとして、一方の鍵を不特定のユーザが使用可能な公開鍵として、他方を秘密に保つ秘密鍵とするものである。例えば、データ暗号化鍵を公開鍵とし、復号鍵を秘密鍵とする。あるいは、認証子生成鍵を秘密鍵とし、認証子検証鍵を公開鍵とする等の態様において使用される。

【０１３５】暗号化、復号化に共通の鍵を用いるいわゆる共通鍵暗号化方式と異なり、公開鍵暗号方式では秘密に保つ必要のある秘密鍵は、特定の１ユーザが持てばよいため鍵の管理において有利である。ただし、公開鍵暗号方式は共通鍵暗号化方式に比較してデータ処理速度が遅く、秘密鍵の配送、ディジタル署名等のデータ量の少ない対象に多く用いられている。公開鍵暗号方式の代表

的なものにはＲＳＡ（Rivest-Shamir-Adleman）暗号がある。これは非常に大きな２つの素数（例えば１５０桁）の積を用いるものであり、大きな２つの素数（例えば１５０桁）の積の素因数分解する処理の困難さを利用している。

【０１３６】公開鍵暗号方式では、不特定多数に公開鍵を使用可能とする構成であり、配布する公開鍵が正当なものであるか否かを証明する証明書、いわゆる公開鍵証明書を使用する方法が多く用いられている。例えば、利用者Ａが公開鍵、秘密鍵のペアを生成して、生成した公開鍵を認証局に対して送付して公開鍵証明書を認証局から入手する。利用者Ａは公開鍵証明書を一般に公開する。不特定のユーザは公開鍵証明書から所定の手続きを経て公開鍵を入手して文書等を暗号化して利用者Ａに送付する。利用者Ａは秘密鍵を用いて暗号化文書等を復号する等のシステムである。また、利用者Ａは、秘密鍵を用いて文書等に署名を付け、不特定のユーザが公開鍵証明書から所定の手続きを経て公開鍵を入手して、その署名の検証を行なうシステムである。

【０１３７】公開鍵証明書は、公開鍵暗号方式における認証局（ＣＡ：Certificate Authority）が発行する証明書であり、ユーザが自己のＩＤ、公開鍵等を認証局に提出することにより、認証局側が認証局のＩＤや有効期限等の情報を付加し、さらに認証局による署名を付加して作成される証明書である。

【０１３８】図１１に公開鍵証明書のフォーマットの概略を示す。各データの概要について説明する。証明書のバージョン（version）番号は、公開鍵証明書フォーマットのバージョンを示す。証明書の通し番号は、シリアルナンバ（ＳＮ：Serial Number）であり、公開鍵証明書発行局（認証局：ＣＡ）によって設定される公開鍵証明書のシリアルナンバである。署名アルゴリズム識別子フィールド（Signature algorithm Identifier）の署名アルゴリズム（algorithm）、アルゴリズムパラメータ（parameters）は、公開鍵証明書の署名アルゴリズムとそのパラメータを記録するフィールドである。なお、署名アルゴリズムとしては、楕円曲線暗号およびＲＳＡがあり、楕円曲線暗号が適用されている場合はパラメータおよび鍵長が記録され、ＲＳＡが適用されている場合には鍵長が記録される。発行局（認証局：ＣＡ）の名前は、公開鍵証明書の発行者、すなわち公開鍵証明書発行局（ＣＡ）の名称（Issuer）が識別可能な形式（Distinguished Name）で記録されるフィールドである。証明書の有効期限(validity)は、証明書の有効期限である開始日時、終了日時が記録される。公開鍵証明書の利用者名（Subject）は、ユーザである認証対象者の識別データが記録される。具体的には例えばユーザ機器のＩＤや、サービス提供主体のＩＤ等の識別子またはカテゴリが記録される。利用者公開鍵フィールド（subject Public Key Info）の鍵アルゴリズム（algorithm）と鍵（subject Public key）は、ユーザの公開鍵情報としての鍵アルゴリズム、鍵情報そのものを格納するフィールドである。オプション領域には、ユーザの属性データ、その他公開鍵証明書の発行、利用に伴うオプショナルデータを記録する。属性データとしては、ユーザの所属グループ情報としてのデバイスマネージャコード（ＤＭＣ）、パーティションマネージャコード（ＰＭＣ）が記録される。なお、ここでユーザは公開鍵証明書のユーザであり、例えばデバイスマネージャ、パーティションマネージャ、チケットユーザ、チケット発行手段、デバイスなどである。

【０１３９】オプション領域には、さらにカテゴリ情報として、チケットユーザ、チケット発行手段、デバイス、デバイスマネージャ、パーティションマネージャなどのエンテイティ、機器種別を示すカテゴリが記録される。

【０１４０】なお、デバイスマネージャがパーティション登録チケット発行手段（PRT Issuer）を兼ねる場合は、後述するパーティション登録チケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）は、デバイスマネージャコード（ＤＭＣ）として設定可能であり、また、パーティションマネージャがファイル登録チケット発行手段、サービス許可チケット発行手段を兼ねる場合は、ファイル登録チケット発行手段コード（ＦＲＴＩＣ：FRT Issuer Code）、サービス許可チケット発行手段コード（ＳＰＴＩＣ：SPT Issuer Code）をパーティションマネージャコード（ＰＭＣ）として設定可能である。なお、これらのコードは後述する各チケット（ＰＲＴ，ＦＲＴ，ＳＰＴなど）にも記録される。

【０１４１】また、各チケット発行手段にデバイスマネージャコード（ＤＭＣ）、パーティションマネージャコード（ＰＭＣ）と異なる独自のコードを割り当てる構成としてもよい。この場合のコード付与は、前述のコード管理機関が実行する。

【０１４２】発行局署名は、公開鍵証明書発行局（ＣＡ）の秘密鍵を用いて公開鍵証明書のデータに対して実行される電子署名であり、公開鍵証明書の利用者は、公開鍵証明書発行局（ＣＡ）の公開鍵を用いて検証を行ない、公開鍵証明書の改竄有無がチェック可能となっている。

【０１４３】公開鍵暗号方式を用いた電子署名の生成方法について、図１２を用いて説明する。図１２に示す処理は、ＥＣ－ＤＳＡ（（Elliptic Curve Digital SignatureAlgorithm）、IEEE P1363/D3）を用いた電子署名データの生成処理フローである。なお、ここでは公開鍵暗号として楕円曲線暗号（Elliptic Curve Cryptography（以下、ＥＣＣと呼ぶ））を用いた例を説明する。なお、本発明のデータ処理装置においては、楕円曲線暗号以外にも、同様の公開鍵暗号方式における、例えばＲＳＡ暗号（（Rivest、Shamir、Adleman）など（ANSI X9.3

1））を用いることも可能である。

【０１４４】図１２の各ステップについて説明する。ステップＳ１において、ｐを標数、ａ、ｂを楕円曲線の係数（楕円曲線：$y^2=x^3+ax+b$，$4a^3+27b^2\neq 0$（ｍｏｄ　ｐ））、Ｇを楕円曲線上のベースポイント、ｒをＧの位数、Ｋｓを秘密鍵（０＜Ｋｓ＜ｒ）とする。ステップＳ２おいて、メッセージＭのハッシュ値を計算し、f=Hash(M)とする。

【０１４５】ここで、ハッシュ関数を用いてハッシュ値を求める方法を説明する。ハッシュ関数とは、メッセージを入力とし、これを所定のビット長のデータに圧縮し、ハッシュ値として出力する関数である。ハッシュ関数は、ハッシュ値（出力）から入力を予測することが難しく、ハッシュ関数に入力されたデータの1ビットが変化したとき、ハッシュ値の多くのビットが変化し、また、同一のハッシュ値を持つ異なる入力データを探し出すことが困難である特徴を有する。ハッシュ関数としては、ＭＤ４、ＭＤ５、ＳＨＡ－１などが用いられる場合もあるし、ＤＥＳ－ＣＢＣが用いられる場合もある。この場合は、最終出力値となるＭＡＣ（チェック値：ＩＣＶに相当する）がハッシュ値となる。

【０１４６】続けて、ステップＳ３で、乱数ｕ（０＜ｕ＜ｒ）を生成し、ステップＳ４でベースポイントをｕ倍した座標Ｖ（Ｘｖ，Ｙｖ）を計算する。なお、楕円曲線上の加算、２倍算は次のように定義されている。

【０１４７】

【数１】P=(Ｘａ，Ｙａ),Q=(Ｘｂ，Ｙｂ),R=(Ｘｃ,Ｙｃ)=P+Qとすると、P≠Qの時（加算）、

$Xc=\lambda^2-Xa-Xb$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(Yb-Ya)/(Xb-Xa)$

P=Qの時（2倍算）、

$Xc=\lambda^2-2Xa$

$Yc=\lambda\times(Xa-Xc)-Ya$

$\lambda=(3(Xa)^2+a)/(2Ya)$

【０１４８】これらを用いて点Ｇのｕ倍を計算する（速度は遅いが、最もわかりやすい演算方法として次のように行う。Ｇ、２×Ｇ、４×Ｇ・・を計算し、uを2進数展開して1が立っているところに対応する$2^i\times G$（Ｇをｉ回２倍算した値（ｉはｕのＬＳＢから数えた時のビット位置））を加算する。

【０１４９】ステップＳ５で、ｃ＝Ｘｖｍｏｄ　ｒを計算し、ステップＳ６でこの値が０になるかどうか判定し、０でなければステップＳ７でｄ＝［（ｆ＋ｃＫｓ）／ｕ］ｍｏｄ　ｒを計算し、ステップＳ８でｄが０であるかどうか判定し、ｄが０でなければ、ステップＳ９でｃおよびｄを電子署名データとして出力する。仮に、ｒを１６０ビット長の長さであると仮定すると、電子署名データは３２０ビット長となる。

【０１５０】ステップＳ６において、ｃが０であった場合、ステップＳ３に戻って新たな乱数を生成し直す。同様に、ステップＳ８でｄが０であった場合も、ステップＳ３に戻って乱数を生成し直す。

【０１５１】次に、公開鍵暗号方式を用いた電子署名の検証方法を、図１３を用いて説明する。ステップＳ１１で、Ｍをメッセージ、ｐを標数、ａ、ｂを楕円曲線の係数（楕円曲線：$y^2=x^3+ax+b$，$4a^3+27b^2\neq 0$（ｍｏｄ　ｐ））、Ｇを楕円曲線上のベースポイント、ｒをＧの位数、ＧおよびＫｓ×Ｇを公開鍵（０＜Ｋｓ＜ｒ）とする。ステップＳ１２で電子署名データｃおよびｄが０＜ｃ＜ｒ、０＜ｄ＜ｒを満たすか検証する。これを満たしていた場合、ステップＳ１３で、メッセージＭのハッシュ値を計算し、ｆ＝Ｈａｓｈ（Ｍ）とする。次に、ステップＳ１４でh=1/d mod rを計算し、ステップＳ１５でｈ１＝ｆｈ　ｍｏｄ　ｒ、ｈ２＝ｃｈｍｏｄ　ｒを計算する。

【０１５２】ステップＳ１６において、既に計算したｈ１およびｈ２を用い、点Ｐ＝（Ｘｐ，Ｙｐ）＝ｈ１×Ｇ＋ｈ２・Ｋｓ×Ｇを計算する。電子署名検証者は、ベースポイントＧおよびＫｓ×Ｇを知っているので、図１２のステップＳ４と同様に楕円曲線上の点のスカラー倍の計算ができる。そして、ステップＳ１７で点Ｐが無限遠点かどうか判定し、無限遠点でなければステップＳ１８に進む（実際には、無限遠点の判定はステップＳ１６でできてしまう。つまり、Ｐ＝（Ｘ，Ｙ）、Ｑ＝（Ｘ，－Ｙ）の加算を行うと、λが計算できず、Ｐ＋Ｑが無限遠点であることが判明している）。ステップＳ１８でＸｐ　ｍｏｄ　ｒを計算し、電子署名データｃと比較する。最後に、この値が一致していた場合、ステップＳ１９に進み、電子署名が正しいと判定する。

【０１５３】電子署名が正しいと判定された場合、データは改竄されておらず、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したことがわかる。

【０１５４】ステップＳ１２において、電子署名データｃまたはｄが、０＜ｃ＜ｒ、０＜ｄ＜ｒを満たさなかった場合、ステップＳ２０に進む。また、ステップＳ１７において、点Ｐが無限遠点であった場合もステップＳ２０に進む。さらにまた、ステップＳ１８において、Ｘｐ　ｍｏｄ　ｒの値が、電子署名データｃと一致していなかった場合にもステップＳ２０に進む。

【０１５５】ステップＳ２０において、電子署名が正しくないと判定された場合、データは改竄されているか、公開鍵に対応した秘密鍵を保持する者が電子署名を生成したのではないことがわかる。

【０１５６】本発明のシステムにおけるデバイスは、デバイスマネージャの管理登録局を介してデバイスに対して発行されるデバイス対応の公開鍵証明書（CERT-DEV）をデバイスに格納し、さらに、パーティションマネージャの管理登録局を介してデバイスのパーティションに対して発行されるパーティション対応の公開鍵証明書（CE

RT-PAR）をデバイスの各パーティションに格納する。これらの公開鍵証明書は、デバイスに対する処理、すなわちパーティション登録チケット（ＰＲＴ）を適用したパーティション登録設定処理、ファイル登録チケット（ＦＲＴ）を適用したファイル登録設定処理、さらにサービス許可チケット（ＳＰＴ）を適用したデータ処理において、チケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）とデバイス間の相互認証、署名生成、検証処理等に適用される。これらの処理の具体例については、後段でフローを用いて説明する。

【０１５７】［Ａ７．デバイスのメモリ部における格納データ］次に、本発明のパーティション分割されたメモリ領域を持つデバイスの格納データについて説明する。先に、図６を用いて説明した通り、デバイスは例えばＥＥＰＲＯＭからなるメモリ部を持ち、主要領域としてパーティション領域、未使用領域、デバイス固有情報およびデバイス内パーティション情報領域を有する。これら各領域の格納データについて以下、図を参照して順次説明する。

【０１５８】（Ａ７．１．デバイス固有情報およびデバイス内パーティション情報領域）まず、デバイス固有情報およびデバイス内パーティション情報領域の各データについて説明する。デバイス固有情報およびデバイス内パーティション情報領域には、デバイス製造エンティティの情報、デバイスマネージャに関する情報、設定パーティション情報、デバイスに対するアクセスを実行してパーティションの設定登録処理を実行する際に必要となる鍵情報などが格納される。

【０１５９】図１４は、製造情報ブロック（Manufacture Information Block）のデータ構成を示している。各領域の数値は、バイト数を示す。図６を用いて説明したように、本実施例の構成では１ブロック：３２バイト構成である。なお、図中グレー部分は暗号化されたデータでも、暗号化されていなくてもよい。

【０１６０】製造情報ブロック（Manufacture Information Block）には、以下のデータが格納される。

* Writable Flag：ブロックへ書き込みが許可されているかの判別フラグ(ex.0xffff：書込許可，others：書込不可)
* Manufacture Code：カード製造工場識別番号
* Manufacture Equipment Code：カード製造ライン番号
* Manufacture Date：カード製造日。例えば、2001年1月1日を0x0000とする
* Manufacture Serial Number：カード製造シリアル番号
* Device Vender Code：ICチップ供給会社番号
* Device Code：ICチップの型番
* Device Parameter：その他のパラメータ

【０１６１】これらのブロックに書かれる情報は一意になるので、これらの情報を基にデバイス（Device）固有識別子としてＩＤｍと定義する。なお、デバイス（Device）固有識別子は製造情報ブロック（Manufacture Information Block）に書き込まれた情報全体、あるいは書き込まれた情報の一部、または書き込まれた情報に基づいて取得される演算データから取得する構成とすることも可能である。

【０１６２】図１５は、デバイス管理情報ブロック（Device Management Information Block）のデータ構成を示す。デバイス管理情報ブロック（Device Management Information Block）には以下のデータが格納される。

【０１６３】* Writable Flag：ブロックへ書き込みが許可されているかの判別フラグ（ex. 0xffff：書込許可，others：書込不可）
* DMC（Device Manager Code）：デバイスマネージャ（ＤＭ：Device Manager）の識別番号
* DMC Version：デバイスマネージャコード（ＤＭＣ）のバージョン。例えば、ＤＭＣを更新する時の比較条件として用いられる。
* Total Block Number in Device：デバイス（Device）内の全ブロック数
* Free Block Number in Device：デバイス（Device）内の空きブロック数
* Partition Number：現在登録されているパーティション（Partition）数
* Pointer of Free Area：空き領域のポインタ

【０１６４】図１６は、公開鍵系デバイス鍵定義ブロック（Device Key Definition Block（PUB））のデータ構成を示す。公開鍵系デバイス鍵定義ブロック（Device KeyDefinition Block（PUB））には以下のデータが格納される。

【０１６５】* PUB_CA(DEV) Pointer：デバイスマネージャの管轄する登録局を介して公開鍵証明書の発行を行なうデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の公開鍵が格納されているブロックへのポインタ
* PUB_CA(DEV) Size：認証局ＣＡ（ＤＥＶ）の公開鍵のサイズ
* PRI_DEV Pointer：デバイス（Device）の秘密鍵が格納されているブロックへのポインタ
* PRI_DEV Size：デバイス（Device）の秘密鍵のサイズ
* PARAM_DEV Pointer：デバイス（Device）の公開鍵パラメータが格納されているブロックへのポインタ
* PARAM_DEV Size：デバイス（Device）の公開鍵パラメータのサイズ
* CERT_DEV Pointer：認証局ＣＡ（ＤＥＶ）が発行したデバイス（Device）の公開鍵証明書が格納されているブロックへのポインタ
* CERT_DEV Size：：認証局ＣＡ（ＤＥＶ）が発行したデバイス（Device）の公開鍵証明書のサイズ
* CRL_DEV Pointer：デバイス（Device）のリボケーシ

ョンリスト（Revocation List）が格納されているブロックへのポインタ
* CRL_DEV Size :デバイス（Device）のリボケーションリスト（Revocation List）のサイズ
* PRTIC（PRT Issuer Category）:パーティション登録チケット（ＰＲＴ）発行者カテゴリ
* PRTIC Version：パーティション登録チケット（ＰＲＴ）発行者カテゴリ（ＰＲＴＩＣ）のバージョン
* DUTIC_DEV（DUT Issuer Category）:データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者カテゴリ
* DUTIC_DEV Version :データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者（ＤＵＴＩＣ）のバージョン

【０１６６】なお、上記のデータ中のリボケーションリストとは、不正デバイスのリストとして、例えばデバイス流通システムの管理者が発行するデバイス排除用リストであり、不正デバイスの識別データをリスト化したデータである。デバイスアクセス機器としてのリーダライタにセットされたデバイスがリボケーションリストに記載されたデバイスである場合は処理を停止するなどの措置をとる。

【０１６７】例えばデバイスに対する処理を実行するすべてのデバイスアクセス機器としてのリーダライタに常に最新のリボケーションリストを配布してデバイスに対して処理を実行する際にリストを参照して処理の実行または停止を判定する。あるいはデバイスアクセス機器としてのリーダライタの通信機能によりネットワークを介して最新のリボケーションリストを閲覧することでリストに記載された不正デバイス情報を取得して処理の実行または停止を判定する。リボケーションリストを利用した具体的処理については、フローを用いた説明中で後述する。

【０１６８】また、上記のデータ中のデータアップデートチケット（ＤＵＴ：Data UpdateTicket）は、デバイスに格納された様々なデータの更新処理を実行する際に更新処理を許可し制限するためのアクセス制限チケットであり、前述のＰＲＴ，ＦＲＴ，ＳＰＴの各チケットと同様、デバイスに対するアクセスルールを記録したチケットである。このデータアップデートチケット（ＤＵＴ：Data Update Ticket）については、後段でさらに詳細に説明する。

【０１６９】図１７は、共通鍵系デバイス鍵定義ブロック（Device Key Definition Block(Common)）のデータ構成を示す。共通鍵系デバイス鍵定義ブロック（Device KeyDefinition Block(Common)）には以下のデータが格納される。

【０１７０】* Mkauth_DEV_A Pointer :双方向個別鍵認証用マスター鍵(MKauth_DEV_A)のポインタ
* Mkauth_DEV_A Size :双方向個別鍵認証用マスター鍵(MKauth_DEV_A)のサイズ
* Kauth_DEV_B Pointer :双方向個別鍵認証用鍵(Kauth_DEV_B)のポインタ
* Kauth_DEV_B Size :双方向個別鍵認証用鍵(Kauth_DEV_B)のサイズ
* Kprt Pointer :パーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵(Kprt)が格納されているブロックへのポインタ
* Kprt Size：パーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵(Kprt)のサイズ
* Kdut_DEV1-4 Pointer :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵(Kdut)が格納されているブロックへのポインタ
* Kdut_DEV1-4 Size：データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵(Kdut)のサイズ
* IRL_DEV Pointer :デバイス（Device）のリボケーションリスト（Revocation List）として、不正デバイスのデバイスＩＤ(Device ID)が格納されているブロックへのポインタ
* IRL_DEV Size :デバイス（Device）のリボケーションリスト（Revocation List）のサイズ

【０１７１】上述のデータ中に示される双方向個別鍵認証の方法、ＭＡＣ（Message Authenticate Code）検証処理については、後段で詳細に説明する。また、Kdut_DEV は４種類存在し、(Kdut_DEV1, Kdut_DEV2)，(Kdut_DEV3, Kdut_DEV4) のペアで使われる。例えば、Kdut_DEV1，3 はMAC生成用、Kdut_DEV2，4は暗号用に使われる。

【０１７２】図１８は、デバイス鍵領域（Device Key Area）のデータ構成を示す。デバイス鍵領域（Device Key Area）には以下のデータが格納される。なお、デバイス鍵領域（Device Key Area）の各格納鍵には、バージョン情報が併せて格納される。鍵の更新時には、バージョンについても併せて更新される。

【０１７３】* IRL_DEV :排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List (Device ID)）
* CRL_DEV :排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)）
* Kdut_DEV1 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_DEV2 :データ更新用暗号鍵
* Kdut_DEV3 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_DEV4 :データ更新用暗号鍵
* Kprt :パーティション登録チケット（ＰＲＴ）のＭＡ

Ｃ検証用鍵
* CERT_DEV ： デバイスマネージャ対応公開鍵を発行する認証局ＣＡ（ＤＥＶ）が発行したデバイス（Device）の公開鍵証明書
* PRI_DEV ：デバイス（Device）の秘密鍵
* PARAM_DEV ：デバイス（Device）の公開鍵パラメータ
* PUB_CA(DEV) ：デバイスマネージャ対応公開鍵を発行する認証局ＣＡ（ＤＥＶ）の公開鍵
* Kauth__DEV_B ：双方向個別鍵認証用共通鍵
* MKauth__DEV_A ：双方向個別鍵認証用マスター鍵

【０１７４】なお、図に示すデバイス鍵領域（Device Key Area）には Kauth__DEV_A ：双方向個別鍵認証用共通鍵、MKauth__DEV_B ：双方向個別鍵認証用マスター鍵が格納されているが、これらの鍵は、デバイスが共通鍵認証処理を行なう要請が無い場合は格納しない構成としてもよく、また、Kprt ：パーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵についても、デバイスがチケット検証処理を実行しない構成の場合には格納しない構成としてもよい。

【０１７５】また、IRL_DEV ：排除デバイス（Device）のデバイス識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（Device ID））、 CRL_DEV ：排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate）、についても、リボーク（排除）されたデバイスが存在しない場合、あるいは他のソースを使用してリボケーションリストを取得する構成とする場合には、リボケーションリストを格納しない構成としてもよい。

【０１７６】図１９は、パーティション定義ブロック（Partition Definition Block）のデータ構成を示す。パーティション定義ブロック（Partition Definition Block）には以下のデータが格納される。

【０１７７】* PMC（Partition Manager Code）：パーティションマネージャ（Partition Manager）に割り当てられたコード（ＰＭＣ）。例えば番号。
* PMC Version ： パーティションマネージャコード（ＰＭＣ）のバージョン
* Partition Start Position ：パーティション（Partition）格納先スタートアドレス
* Partition Size ：パーティション（Partition）のサイズ

【０１７８】以上が、デバイスのメモリ部のデバイス固有情報およびデバイス内パーティション情報領域の各データである。

【０１７９】（Ａ７．２．パーティション領域）次に、パーティション領域の各データについて説明する。パーティション領域は、パーティションマネージャの管理領域である。前述したように、デバイスマネージャの管理するパーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）の発行したＰＲＴチケットに基づいて各サービス主体としてのパーティションマネージャが所定の手続き、すなわちパーティション登録チケット（ＰＲＴ）に設定されたルールに従ってデバイス内のメモリに設定する。以下、パーティション領域のデータ構成について説明する。

【０１８０】図２０は、パーティション管理情報ブロック（Partition Management Information Block）のデータ構成を示す。パーティション管理情報ブロック（Partition Management Information Block）には以下のデータが格納される。

【０１８１】* PMC（Partition Manager Code）：パーティション（Partition）保有者の番号
* PMC Version ：パーティションマネージャコード（ＰＭＣ）のバージョン
* Total Block Number in Partition ：パーティション（Partition）内の全ブロック数
* Free Block Number in Partition ： パーティション（Partition）内の空きブロック数
* Pointer of Free Area ： パーティション（Partition）内の未使用領域のポインタ
* File Number ：パーティションに現在登録されているファイル（File）数

【０１８２】図２１は、公開鍵系パーティション鍵情報ブロック（Partition Key Definition Block(PUB)）のデータ構成を示す。公開鍵系パーティション鍵情報ブロック（Partition Key Definition Block(PUB)）には以下のデータが格納される。

【０１８３】* PUB_CA(PAR) Pointer ：パーティションマネージャの管轄登録局を介して公開鍵証明書を発行する認証局ＣＡ（ＰＡＲ）の公開鍵が格納されているブロックへのポインタ
* PUB_CA(PAR) Size ：認証局ＣＡ（ＰＡＲ）の公開鍵のサイズ
* PRI_PAR Pointer ：パーティション（Partition）の秘密鍵が格納されているブロックへのポインタ
* PRI_PAR Size ：パーティション（Partition）の秘密鍵のサイズ
* PARAM_PAR Pointer ：パーティション（Partition）の公開鍵パラメータが格納されているブロックへのポインタ
* PARAM_PAR Size ：パーティション（Partition）の公開鍵パラメータのサイズ
* CERT_PAR Pointer ：認証局ＣＡ（ＰＡＲ）が発行したパーティション（Partition）の公開鍵証明書が格納されているブロックへのポインタ
* CERT_PAR Size ：認証局ＣＡ（ＰＡＲ）が発行したパーティション（Partition）の公開鍵証明書のサイズ
* CRL_PAR Pointer ：パーティション（Partition）のリボケーションリスト（Revocation List）が格納されて

いるブロックへのポインタ
* CRL_PAR Size :パーティション（Partition）のリボケーションリスト（Revocation List）のサイズ
* FRTIC（FRT Issuer Category）:ファイル登録チケット（ＦＲＴ）発行者カテゴリ
* FRTIC Version : ファイル登録チケット（ＦＲＴ）発行者カテゴリ（ＦＲＴＩＣ）のバージョン
* DUTIC_PAR（DUT Issuer Category）:データアップデートチケット（ＤＵＴ）発行者カテゴリ
* DUTIC_PAR Version : データアップデートチケット（ＤＵＴ）発行者カテゴリ（ＤＵＴＩＣ）のバージョン

【０１８４】図２２は、共通鍵系パーティション鍵情報ブロック（Partition Key Definition Block(Common)）データ構成を示す。共通鍵系パーティション鍵情報ブロック（Partition Key Definition Block(Common)）には以下のデータが格納される。

【０１８５】* Mkauth_PAR_A Pointer :双方向個別鍵認証用マスター鍵(MKauth_PAR_A)のポインタ
* Mkauth_PAR_A Size :双方向個別鍵認証用マスター鍵(MKauth_PAR_A)のサイズ
* Kauth_PAR_B Pointer :双方向個別鍵認証用鍵(Kauth_PAR_B)のポインタ
* Kauth_PAR_B Size :双方向個別鍵認証用鍵(Kauth_PAR_B)のサイズ
* Kfrt Pointer :ファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵(Kfrt)が格納されているブロックへのポインタ
* Kfrt Size :ファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵(Kfrt)のサイズ
* Kdut_PAR1-4 Pointer :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵(Kdut)が格納されているブロックへのポインタ
* Kdut_PAR1-4 Size : データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵(Kdut)のサイズ
* IRL_PAR Pointer :パーティション（Partition）の排除デバイスのＩＤを格納したリボケーションリスト（Revocation List－Device ID)が格納されているブロックへのポインタ
* IRL_PAR Size :パーティション（Partition）のリボケーションリスト（Revocation List－Device ID)のサイズ

【０１８６】上述のデータ中に示される双方向個別鍵認証の方法、ＭＡＣ（Message Authenticate Code）検証処理については、後段で詳細に説明する。また、Kdut_PAR は４種類存在し、(Kdut_PAR1, Kdut_PAR2), (Kdut_PAR3, Kdut_PAR4) のペアで使われる。例えば、Kdut_PAR 1, 3 はMAC生成用、Kdut_PAR2, 4は暗号用に使われる。

【０１８７】図２３は、パーティション鍵領域（Partition Key Area）のデータ構成を示す。パーティション鍵領域（Partition Key Area）には以下のデータが格納される。なお、パーティション鍵領域（Partition Key Area）の各格納鍵には、バージョン情報が併せて格納される。鍵の更新時には、バージョンについても併せて更新される。

【０１８８】* IRL_PAR :パーティションアクセス排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト(Revocation List (Device ID))
* CRL_PAR :パーティションアクセス排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)
* Kdut_PAR1 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_PAR2 :データ更新用暗号鍵
* Kdut_PAR3 : データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_PAR4 :データ更新用暗号鍵
* Kfrt :ファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵
* CERT_PAR :認証局ＣＡ（ＰＡＲ）が発行したパーティション（Partition）の公開鍵証明書
* PRI_PAR :パーティション（Partition）の秘密鍵
* PARAM_PAR :パーティション（Partition）の公開鍵パラメータ
* PUB_CA(PAR) :認証局ＣＡ（ＰＡＲ）の公開鍵
* Mkauth_PAR_A :双方向個別鍵認証用マスター鍵
* Kauth_PAR_B :双方向個別鍵認証用共通鍵

【０１８９】図２４は、ファイル定義ブロック（ＦＤＢ：File Definition Block）のデータ構成を示す。ファイル定義ブロック（File Definition Block）には以下のデータが格納される。

【０１９０】* File ID :ファイル（File）識別名
* File Start Position :ファイル（File）スタートアドレス
* File Size :ファイル（File）サイズ
* SPTIC（SPT Issuer Category）:サービス許可チケット（ＳＰＴ）発行者カテゴリ
* SPTIC Version : サービス許可チケット（ＳＰＴ）発行者カテゴリ（SPTIC）のバージョン
* File Structure Type Code :ファイル構造タイプ（File Structure Type）のコード
* Acceptable Authentication Type :許容認証タイプを示す。各ファイル構造タイプ（File Structure Type）に対して定義されるアクセスモードとこのフィールドの各ビット（本例では最大１６個）が対応する。詳細は下記に説明する。

* Acceptable Verification Type:許容検証タイプを示す。各ファイル構造タイプ（File Structure Type）に対して、定義されるアクセスモードとこのフィールドの各ビット（本例では最大１６個）が対応する。詳細は下記に説明する。

* Kspt ：サービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵(Kspt)

【０１９１】上記の許容認証タイプ（Acceptable Authentication Type）は、各ファイル構造タイプ（File Structure Type）に対して定義されるアクセスモードとこのフィールドの各ビット（本例では最大１６個）が対応するように設定された許容認証タイプであり、例えばあるアクセスモードを実行する際に、そのモードに対応するビットに１がたっている場合には、公開鍵認証が済んで認証済みでないと実行されれないものとする。これにより、より重要度の高いコマンド（例えば入金処理など）の実行の際には、公開鍵認証を義務づけ、安全性を確保できる。チケットを用いることで同様の制御も可能ではあるが、許容認証タイプ（AcceptableAuthentication Type）は、チケットと異なり、ファイル定義ブロック（ＦＤＢ：File Definition Block）の一部としてデバイスに格納されることになるため、この情報はファイル生成後に変更されることがない。従って、絶対に許容認証タイプの変更を許さない強い制約を与えたいときに利用することにより、安全性の最低限の保証を与えることができる。

【０１９２】また上記の許容検証タイプ（Acceptable Verification Type）は、各ファイル構造タイプ（File Structure Type）に対して定義されるアクセスモードとこのフィールドの各ビット（本例では最大１６個）が対応するように設定された許容検証タイプであり、例えばあるアクセスモードを実行する際に、そのモードに対応するビットに１がたっている場合には、公開鍵方式によるチケット検証が済んでないと実行されれないものとする。この例では、各フィールドを２バイトづつにしたため、最大１６個のアクセスモードとの対応付けしかできないが、必要に応じてフィールドサイズを大きくとることにより、より多くのコマンドに対応付ける構成とすることができる。

【０１９３】また、本実施例構成においては、許容認証タイプ（Acceptable Authentication Type）、許容検証タイプ（Acceptable Verification Type）は設定が「１」のときに公開鍵方式の認証または検証を必要とする設定としてあるが、これらの各フィールドを２ビット単位の構成として、値が「１１」の場合には公開鍵方式、「０１」の場合には共通鍵方式、「００」「１０」の場合には公開鍵方式、共通鍵方式のいずれでもは許容する、などの細分化した設定としてもよい。

【０１９４】上述のデータ中のファイル構造タイプ（File Structure Type）は、パーティション内に生成されるファイルの構造を示すコードである。ファイル構造とコードの対応の一例を図２５に示す。

【０１９５】ファイル構造には、図２５に示す各種構造（File Structure）があり、それぞれにコード０００１～０００７が割り当てられる。各構造の意味を以下に示す。

【０１９６】* Random ：本ファイル構造を持つデータはすべての読書きがランダムに可能なファイルである。

* Purse ：本ファイル構造を持つデータは、金額情報データであり、減算（Sub）、加算（Add）など金額の価値変更処理が可能であるデータファイルである。

* Cyclic ：本ファイル構造を持つデータは循環型（Cyclic）のデータ書き込みが可能なファイル構造である。

* Log ：本ファイル構造を持つデータは、ログデータファイルであり、各データ処理情報についての記録情報ファイルである。

* Key ：本ファイル構造を持つデータファイルは、鍵情報ファイルであることを示す。

* 複合ファイル ：上記各種ファイル構造の複合構造（Ｅｘ．PurseとLog）を持つファイルである。複合ファイルには、その組み合わせパターンにより異なるコード（図では０００６：複合ファイル１、０００７：複合ファイル２）が割り当てられる。

【０１９７】以上、デバイスのメモリ部に格納されるデータについて説明した。これらのデータを用いた具体的な処理については、後段で説明する。

【０１９８】［Ａ８．各チケットのデータフォーマット］前述したように、デバイスに対するパーティションの設定登録処理には、正当なチケット発行手段（Ticket Issuer）の発行したパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）、デバイスに設定されたパーティション内に対するファイルの設定登録処理には、正当なチケット発行手段（Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ：File Registration Ticket）、また、各ファイルに対するアクセスには正当なチケット発行手段（Ticket Issuer）の発行したサービス許可チケット（ＳＰＴ：Service Permission Ticket）が必要となる。また、前述のデバイスのメモリ部のデータ説明の欄で簡単に説明したように、デバイス格納データの更新処理にはデータアップデートチケット（ＤＵＴ）を必要とする。

【０１９９】これらの各チケットはデバイスに対するアクセスルールをバイナリデータとして記述したデータ列によって構成される。チケットはデバイスに対する処理に応じて、チケットユーザである例えばデバイスアクセス機器としてのリーダライタからデバイスに送信される。チケットを受信したデバイスはチケットの正当性検証処理を実行し、正当性検証に成功した場合、チケットに記録されたルールに従って各種の処理（ｅｘ．パーティション生成、ファイル生成、データアクセス）が実行

される。以下、これらの各チケットのデータフォーマットについて説明する。

【０２００】（Ａ８．１．パーティション登録チケット（ＰＲＴ））パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）は、デバイスに対するパーティションの設定登録処理の際に適用されるアクセスコントロールチケットである。正当なデバイスマネージャ管轄下のチケット発行手段（Ticket Issuer）の発行したＰＲＴを用い、ＰＲＴに記録された手続きに従って、パーティションマネージャの管轄下のチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）によりデバイスにアクセスすることで、ＰＲＴに記録された制限内でパーティションを設定することができる。

【０２０１】図２６にパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）のデータフォーマットを示す。パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）には以下に説明するデータが格納される。

【０２０２】* Ticket Type :チケット（Ticket）の種別。

* Format Version :チケット（Ticket）のフォーマットバージョン

* Ticket Issuer :デバイスマネージャの識別子（＝ＤＭＣ）

* Serial Number :チケット（Ticket）のシリアル番号

* Size of Ticket :チケット（Ticket）のサイズ

* Authentication Flag :チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

* Ticket User の所属(Group) :チケット（Ticket）利用者の所属

* Authentication Type :デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））

* Ticket User の識別子 :チケット（Ticket）利用者を判別する識別データ（カテゴリまたは識別子）

当フィールドは、［Authentication Type］と連携したデータとされ、［Authentication Type］が公開鍵認証の場合：識別名（ＤＮ：Distinguished Name）またはカテゴリ（Category）またはシリアル番号（ＳＮ）が格納され,共通鍵認証の場合、：認証ＩＤが格納される。認証不要の場合は格納は必須ではない。

* PMC:パーティションマネージャコード（Partition Manager Code）として、パーティション定義ブロック（Partition Definition Block）に記述されるコード

* PMC Version : パーティションマネージャコード（ＰＭＣ）のバージョン

* Operation Type :パーティション（Partition）作成か削除かの指定（作成（Generate） / 削除（Delete））

* Partition Size :パーティション（Partition）のサイズ

* Integrity Check Type :チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public) /共通鍵方式(Common)）

* Integrity Check Value :チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）

【０２０３】なお、パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の公開鍵証明書（CERT_PRTI）も一緒に送信する。ＰＲＴ発行手段の公開鍵証明書（CERT_PRTI）の属性（Attribute）は、ＰＲＴ発行手段（ＰＲＴ Issuer）の識別子(PRTIC)と一致する。

【０２０４】デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録した［Authentication Type］には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２０５】チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）を記録する［Integrity Check Value］フィールドには、公開鍵方式であれば、パーティション登録チケット発行手段（ＰＲＴ Issuer）の秘密鍵に基づく署名（図１２参照）が生成され格納される。デバイスマネージャ自体がパーティション登録チケット発行手段（ＰＲＴ Issuer）を兼ねる場合は、デバイスマネージャの秘密鍵を用いて署名が生成される。署名検証処理（図１３参照）の際は、対応のＣＡ（ＤＥＶ）の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってパーティション登録チケット発行手段（ＰＲＴ Issuer）（ｅｘ．デバイスマネージャ）の公開鍵（公開鍵証明書）を取得することが必要である。

【０２０６】パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の公開鍵証明書（CERT_PRTI）の検証の後、公開鍵証明書（CERT_PRTI）から取り出したパーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の公開鍵によりＩＣＶ（Integrity Check Value）の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【０２０７】（Ａ８．２．ファイル登録チケット（ＦＲＴ））ファイル登録チケット（ＦＲＴ：File Registration Ticket）は、デバイスに対して設定されたパーティ

ションにファイルを設定登録する際に適用されるアクセスコントロールチケットである。正当なパーティションマネージャ管轄下のチケット発行手段（Ticket Issuer）の発行したＦＲＴを用い、ＦＲＴに記録された手続きに従ってチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）によりデバイスにアクセスすることで、ＦＲＴに記録された制限内でファイルを設定することができる。

【０２０８】図２７にファイル登録チケット（ＦＲＴ：File Registration Ticket）のデータフォーマットを示す。ファイル登録チケット（ＦＲＴ：File Registration Ticket）には以下に説明するデータが格納される。

【０２０９】* Ticket Type :チケット（Ticket）の種別

* Format Version :チケット（Ticket）のフォーマットバージョン

* Ticket Issuer :パーティションマネージャの識別子（＝ＰＭＣ）

* Serial Number :チケット（Ticket）のシリアル番号

* Size of Ticket :チケット（Ticket）のサイズ

* Authentication Flag :チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

* Ticket User の所属(Group) :チケット（Ticket）利用者の所属

* Authentication Type :デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））

* Ticket User の識別子 :チケット（Ticket）利用者を判別する識別データ（カテゴリまたは識別子）

当フィールドは、［Authentication Type］と連携したデータとされ、［Authentication Type］が公開鍵認証の場合：識別名（ＤＮ：Distinguished Name）またはカテゴリ（Category）またはシリアル番号（ＣＮ）が格納され，共通鍵認証の場合、：認証ＩＤが格納される。認証不要の場合は格納は必須ではない。

* SPTIC :サービス許可チケット発行手段のコード

* SPTIC Ver : サービス許可チケット発行手段のコード（SPTIC）のバージョン

* File ID :パーティション内に生成するファイル（File）の識別子（ＩＤ）

* Operation Type :ファイルの作成か削除かの指定（生成（Generate）/削除（Delete））

* File Size :生成するファイル（File）のサイズ

* File Structure :生成するファイル（File）のファイル構造（Structure）

* Acceptable Authentication Type :このチケットで定義されるファイルに対するアクセスモードを実行するタめに必要とする相互認証の種類（公開鍵方式、公開鍵、共通鍵いずれでも可）を表すビット列

* Acceptable Verification Type :このチケットで定義されるファイルに対するアクセスモードを実行するタめに必要とするサービス許可チケット（ＳＰＴ）の検証の種類（公開鍵方式、公開鍵、共通鍵いずれでも可）を表すビット列

* Kspt_Encrypted :ファイル定義ブロック（File Definition Block）に記載されるサービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵 Kspt を そのパーティションのファイル登録チケットのＭＡＣ検証用鍵 Kfrt で暗号化したデータKfrt（Kspt）

* Integrity Check Type :チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)）

* Integrity Check Value :チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）

【０２１０】なお、ファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）の公開鍵証明書（CERT_FRTI）も一緒に送信する。ＦＲＴ発行手段の公開鍵証明書（CERT_FRTI）の属性（Attribute）は、ファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）の識別子(FRTIC)と一致する。

【０２１１】デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録した［Authentication Type］には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２１２】チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）を記録する［Integrity Check Value］フィールドには、公開鍵方式であれば、ファイル登録チケット発行手段（ＦＲＴ Issuer）の秘密鍵に基づく署名（図１２参照）が生成され格納される。パーティションマネージャ自体がファイル登録チケット発行手段（ＦＲＴ issuer）を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。署名検証処理（図１３参照）の際は、ファイル登録チケット発行手段の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってファイル登録チケット発行手段（ＦＲＴ Issuer）（ｅｘ．パーティションマネージャ）の公開鍵（公開鍵証明書）を取得することが必要である。

【０２１３】ファイル登録チケット（ＦＲＴ）発行手段

(FRT Issuer) の公開鍵証明書 (CERT_FRTI) の検証の後、公開鍵証明書 (CERT_FRTI) から取り出したファイル登録チケット（ＦＲＴ）発行手段 (FRT Issuer) の公開鍵によりＩＣＶ (IntegrityCheck Value) の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【０２１４】（Ａ８．３．サービス許可チケット（ＳＰＴ））サービス許可チケット（ＳＰＴ：Service Permission Ticket）は、デバイスに対して設定されたパーティション内の各データに対してアクセスしてデータ読み出し、データ書き込み、金額データの減算、加算などの処理を実行する際に適用されるアクセスコントロールチケットである。正当なパーティションマネージャ管轄下のチケット発行手段 (Ticket Issuer) の発行したＳＰＴを用い、ＳＰＴに記録された手続きに従ってチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）によりデバイスにアクセスすることで、ＳＰＴに記録された制限内でデータ処理を実行することができる。

【０２１５】なお、サービス許可チケット（ＳＰＴ：Service Permission Ticket）は、パーティションに設定されたファイルの中から唯一のファイルに対してのみアクセスを許可する形式と、複数のファイルに対するアクセスを許可する形式とがあり、それぞれの形式について説明する。

【０２１６】図２８に、パーティションに設定されたファイルの中から唯一のファイルに対してのみアクセスを許可する形式のサービス許可チケット（ＳＰＴ：Service Permission Ticket）のデータフォーマットを示す。サービス許可チケット（ＳＰＴ：Service Permission Ticket）には以下に説明するデータが格納される。

【０２１７】* Ticket Type :チケット (Ticket) の種別。

* Format Version :チケット (Ticket) のフォーマットバージョン

* Ticket Issuer :パーティションマネージャの識別子（＝ＰＭＣ）

* Serial Number :チケット (Ticket) のシリアル番号

* Size of Ticket :チケット (Ticket) のサイズ

* Authentication Flag :チケット (Ticket) の利用処理においてデバイス (Device) との相互認証が必要か否かを示すフラグ

* Ticket User の所属(Group) :チケット (Ticket) 利用者の所属

* Authentication Type :デバイス (Device) の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)）

* Ticket User の識別子 :チケット (Ticket) 利用者を判別する識別データ（カテゴリまたは識別子）

当フィールドは、[Authentication Type] と連携したデータとされ、[Authentication Type] が公開鍵認証の場合：識別名（ＤＮ：Distinguished Name）またはカテゴリ (Category) またはシリアル番号（ＣＮ）が格納され，共通鍵認証の場合、：認証ＩＤが格納される。認証不要の場合は格納は必須ではない。

* File ID :パーティション内のアクセスファイル (File) の識別子（ＩＤ）

* File Access Mode :アクセスを許諾するファイル (File) へのアクセスモード (Access Mode)

* Integrity Check Type :チケット (Ticket) の正当性検証値の種別（公開鍵方式(Public) /共通鍵方式(Common)）

* Integrity Check Value :チケット (Ticket) の正当性検証値（公開鍵方式：署名 (Signature)、共通鍵方式：ＭＡＣ）

【０２１８】なお、サービス許可チケット（ＳＰＴ）発行手段 (SPT Issuer) の発行したチケット (Ticket) を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット（ＳＰＴ）発行手段 (SPT Issuer) の公開鍵証明書 (CERT_SPTI) も一緒に送信する。ＳＰＴ発行手段の公開鍵証明書 (CERT_SPTI) の属性 (Attribute) は、（ＳＰＴ）発行手段 (SPT Issuer) の識別子(SPTIC)と一致する。

【０２１９】サービス許可チケット（ＳＰＴ）発行手段 (SPT Issuer) を、パーティションマネージャが兼ねる構成においては、サービス許可チケット（ＳＰＴ）発行手段 (SPT Issuer) のコードは、パーティションマネージャコード（ＰＭＣ）として設定することが可能である。

【０２２０】デバイス (Device) の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可 (Any)）を記録した [Authentication Type] には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２２１】チケット (Ticket) の正当性検証値（公開鍵方式：署名 (Signature)、共通鍵方式：ＭＡＣ）を記録する [Integrity Check Value] フィールドには、公開鍵方式であれば、サービス許可チケット発行手段（ＳＰＴ Issuer）の秘密鍵に基づく署名（図１２参照）が生成され格納される。パーティションマネージャ自体がサービス許可チケット発行手段（ＳＰＴ Issuer）を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。署名検証処理（図１３参照）の際は、サービス許可チケット（ＳＰＴ）発行手段 (SPT Issuer) の公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、ま

たは前もってサービス許可チケット発行手段（ＳＰＴ Issuer）（ｅｘ．パーティションマネージャ）の公開鍵（公開鍵証明書）を取得することが必要である。

【０２２２】サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵証明書（CERT_SPTI）の検証の後、公開鍵証明書（CERT_SPTI）から取り出したサービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵によりＩＣＶ（IntegrityCheck Value）の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【０２２３】上述のチケット・フォーマットの説明中、File Access Mode：アクセスを許諾するファイル（File）へのアクセスモード（Access Mode）に記録されるモードとアクセス態様について、図２９を用いて説明する。

【０２２４】ファイルとして生成されるデータは、ユーザの識別データ、金額データ、暗号処理鍵データ、ログデータ、あるいは複合ファイルデータなど様々であり、各データに応じたアクセス処理、すなわちデータ読み取り、書き込み、消去、加算、減算、暗号化、復号…がアクセスデータに対して実行されることになる。

【０２２５】サービス許可チケット（ＳＰＴ）のFile Access Modeには、このような様々なアクセスの態様中、いずれのアクセスモードを許可しているものかを定義している。アクセスモードの一覧を図２９に示す。図２９に示すアクセスモードは一例であり、この他にもデバイスに格納されるデータに応じたアクセスモードを設定することができる。

【０２２６】サービス許可チケット（ＳＰＴ）に設定された［File ID：パーティション内のアクセスファイル（File）の識別子（ＩＤ）］によって示されるファイルに対してファイルアクセスモード［File Access Mode］に定義された処理が実行できる。サービス許可チケット（ＳＰＴ）に設定されたファイルアクセスモードが読み出し（Read）であればファイル内データの読み出しが実行できる。サービス許可チケット（ＳＰＴ）に設定されたファイルアクセスモードが書き込み（Write）であればファイル内へのデータの書き込みが実行できる。

【０２２７】このようなアクセスモードは、前述したファイル構造（File Structure）（図２５参照）によって設定可能なモードが制限される。例えばファイル構造がpurseであれば金額データであり、加算（add）、減算（Sub）等のアクセスモードの設定が可能となる。このようなファイル構造と、設定可能なアクセスモード、さらに、デバイスアクセス機器としてのリーダライタからデバイスに対して送信されるコマンドの例を図３０に示す。

【０２２８】図３０には、ファイル構造がＲａｎｄｏｍの場合と、複合ファイルの場合に設定可能なアクセスモード、およびコマンド例を示している。

【０２２９】例えばファイル構造がＲａｎｄｏｍの場合において、アクセスモードが読み出し（Ｒｅａｄ）である場合、デバイスが許容可能なコマンドは［Ｒｅａｄ］のみとなる。また、同様にファイル構造がＲａｎｄｏｍの場合において、アクセスモードが暗号化読み出し（Ｒｅａｄ）である場合、デバイスが許容可能なコマンドは暗号化読み出し［ＥｎｃＲｅａｄ］のみとなる。

【０２３０】また、ファイル構造がPurseおよびLogを含むような複合ファイルの場合において、アクセスモードが入金系である場合、デバイスが許容可能なコマンドは預け入れ［Ｄｅｐｏｓｉｔ］のみとなる。また、同様にファイル構造がPurseおよびLogを含むような複合ファイルの場合において、アクセスモードが出金系である場合、デバイスが許容可能なコマンドは引き出し［Ｗｉｔｈｄｒａｗ］、レシート生成［Ｍａｋｅ　Ｒｅｃｅｉｐｔ］、レシート読み出し［Ｒｅａｄ　Ｒｅｃｅｉｐｔ］となる。

【０２３１】ファイルアクセスモード（図２９参照）の入金系に対応する許容コマンド（図３０参照）として、上述の預け入れコマンド（Deposit Command）を定義し、アクセス許可チケットのファイルアクセスモード（File Access Mode）に［入金系］を設定し、ファイルＩＤ(File ID)として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット（ＳＰＴ）を生成して、ファイルアクセス装置からデバイスに対して送信し、預け入れコマンド（Deposit Command）とともに、預け入れ金額データを送信することにより、例えば、複合ファイル中のファイル［Ｐｕｒｓｅ］にX円を加算し、複合ファイル中のファイル［Ｌｏｇ］に処理記録を書き込むなどのシーケンシャル処理を実行させることが可能となる。これらの処理についての詳細は、後述する。

【０２３２】図３０に示す他にも、様々なアクセスモード、コマンドの組合わせが可能であり、実際のデバイスの利用形態に応じて設定されることになる。

【０２３３】デバイスは、メモリ部に格納された各ファイルに対して許容されるコマンドの定義データを図３０のようなテーブルとして保有し、前記アクセス機器から入力されるコマンドが前記定義データに定義されたコマンドである場合にのみコマンドを実行する。複合ファイルに対して許容されるコマンドの定義データには、上述したように複合ファイルに含まれる複数ファイルの各々に対応して実行可能な複数のコマンドからなるシーケンスコマンドを含む。

【０２３４】処理対象となる特定のファイルをサービス許可チケット（ＳＰＴ）のファイルＩＤ(File ID)欄に設定し、所定のアクセスモードをサービス許可チケット（ＳＰＴ）のファイルアクセスモード（File Access Mode）に設定することで、当該サービス許可チケット（ＳＰＴ）を利用したファイルアクセスをコントロールする

ことが可能となる。具体的処理については、後段でフローを用いて説明する。

【０２３５】次に、図３１に、パーティションに設定されたファイル中の複数ファイルに対してアクセスを許可する形式のサービス許可チケット（ＳＰＴ：Service Permission Ticket）のデータフォーマットを示す。サービス許可チケット（ＳＰＴ：Service Permission Ticket）には以下に説明するデータが格納される。

【０２３６】* Ticket Type ：チケット（Ticket）の種別。

* Format Version ：チケット（Ticket）のフォーマットバージョン

* Ticket Issuer ：パーティションマネージャの識別子（＝ＰＭＣ）

* Serial Number ：チケット（Ticket）のシリアル番号

* Size of Ticket ：チケット（Ticket）のサイズ

* Authentication Flag ：チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

* Ticket User の所属(Group) ：チケット（Ticket）利用者の所属

* Authentication Type ：デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））

* Ticket User の識別子 ：チケット（Ticket）利用者を判別する識別データ（カテゴリまたは識別子）

当フィールドは、［Authentication Type］と連携したデータとされ、［Authentication Type］が公開鍵認証の場合：識別名（ＤＮ：Distinguished Name）またはカテゴリ（Category）が格納され，共通鍵認証の場合、：認証ＩＤが格納される。認証不要の場合は格納は必須ではない。

* File ID ：パーティション内のアクセスファイル（File）の識別子（ＩＤ）

* File Access Mode ：アクセスを許諾するファイル（File）へのアクセスモード（Access Mode）

* Group of Target File ：アクセスを許すファイル（File）のグループ（Group）

* Target File ID ：アクセスを許すファイル（File）の識別子（ＩＤ）

* Read/Write Permission ： アクセスを許すファイル（File）（ターゲットファイル（Target File））に対する処理態様（読み出し（Read），書き込み（Write））の許可

* Integrity Check Type ：チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)）

* Integrity Check Value ：チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）

【０２３７】このように、Group of Target File ：アクセスを許すファイル（File）のグループ（Group）を定義してチケットに記録することにより、パーティション内の複数のファイルに対するアクセスを唯一のサービス許可チケット（ＳＰＴ）で許可することが可能となる。

【０２３８】なお、上述のサービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際にも、公開鍵方式の場合、サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵証明書（CERT_SPTI）も一緒に送信する。ＳＰＴ発行手段の公開鍵証明書（CERT_SPTI）の属性（Attribute）は、サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の識別子(SPTIC)と一致する。

【０２３９】デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録した［Authentication Type］には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２４０】サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵証明書（CERT_SPTI）の検証の後、公開鍵証明書（CERT_SPTI）から取り出したサービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵によりＩＣＶ（IntegrityCheck Value）の署名検証が可能となる。これらの処理については、フローを用いて後段で説明する。

【０２４１】（Ａ８．４．データアップデートチケット（ＤＵＴ））データアップデートチケット（ＤＵＴ：Data Update Ticket）は、デバイスに格納された様々なデータに対してアクセスしてデータの更新処理を実行する際に適用されるアクセスコントロールチケットである。正当なデータアップデートチケット（ＤＵＴ）発行手段（Ticket Issuer）の発行したＤＵＴを用い、ＤＵＴに記録された手続きに従ってチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）によりデバイスにアクセスすることで、ＤＵＴに記録された制限内でデータ処理を実行することができる。

【０２４２】なお、データアップデートチケット（ＤＵＴ：Data Update Ticket）は、デバイスマネージャの管理するデータ項目の更新処理を実行するために適用されるチケットＤＵＴ（ＤＥＶ）と、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットＤＵＴ（ＰＡＲ）がある。チケット：ＤＵＴ（ＤＥＶ）発行手段はデバイスマネージャの管理下にあり、チケット：ＤＵＴ（ＰＡＲ）発行手段はパーティションマネージャの管理

下にある。

【０２４３】図３２に、２つのデータアップデートチケット（ＤＵＴ：Data Update Ticket）、ＤＵＴ（ＤＥＶ）、ＤＵＴ（ＰＡＲ）のデータフォーマットを示す。データアップデートチケット（ＤＵＴ：Data Update Ticket）には以下に説明するデータが格納される。

【０２４４】＊Ticket Type：チケット（Ticket）の種別（ＤＵＴ（ＤＥＶ）／ＤＵＴ（ＰＡＲ）

＊Format Version：チケット（Ticket）のフォーマットバージョン

＊Ticket Issuer：デバイス／パーティションマネージャの識別子。チケット（Ticket）の種別（Ticket Type）がＤＵＴ（ＤＥＶ）ならＤＭＣ、ＤＵＴ（ＰＡＲ）ならＰＭＣとなる

＊Serial Number：チケット（Ticket）のシリアル番号

＊Size of Ticket：チケット（Ticket）のサイズ

＊Ticket User の所属(Group)：チケット（Ticket）利用者の所属

＊Ticket User の識別子：チケット（Ticket）利用者を判別する識別データ（カテゴリまたは識別子）

当フィールドは、［Authentication Type］と連携したデータとされ、［Authentication Type］が公開鍵認証の場合：識別名（ＤＮ：Distinguished Name）またはカテゴリ（Category）が格納され，共通鍵認証の場合、：認証ＩＤが格納される。認証不要の場合は格納は必須ではない。

＊Authentication Type：デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））

＊Encrypted Flag：更新されるデータが暗号化されているか否か（暗号化：Encrypted /非暗号化：none）

＊Old Data Code：更新される古いデータのコード（Code）

＊Data Version Rule：データ更新をする時のバージョン条件

＊Data Version Condition：データ更新をする時のバージョン値

＊Size of New Data：更新する新しいデータのサイズ

＊New Data：更新する新しいデータ（暗号化される場合もある）。

＊New Data Version：更新するデータのバージョン

＊Integrity Check Type：チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public) /共通鍵方式(Common)）

＊Integrity Check Value：チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）

【０２４５】データアップデートチケット（ＤＵＴ：Data Update Ticket）を適用したデータ更新をする際に、［Data Version Rule：データ更新をする時のバージョン条件］と、［Data Version Condition：データ更新をする時のバージョン値］、これら２つのフィールドの組み合わせにより条件を表現する。

【０２４６】データ更新をする時のバージョン条件［Data Version Rule］は、Any, Exact, Older の３種類が存在する。Any はバージョン（Version）条件に無関係でデータ更新が可能、Exact は、続く［Data Version Condition］に指定された値と同じ場合にデータ更新が可能、Older は、New Data Version の方が新しい場合にのみデータ更新が可能となる。なお、バージョン条件［Data Version Rule］がAny, または Older の場合は、［Data Version Condition］は使用しないかもしくは無視する。

【０２４７】デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録した［Authentication Type］には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２４８】データアップデートチケット-ＤＵＴ（ＤＥＶ）発行手段（DUT Issuer）を、デバイスマネージャが兼ねる構成においては、データアップデートチケット－ＤＵＴ（ＤＥＶ）発行手段（DUT Issuer）のコード（チケットユーザ（Ticket User））は、デバイスマネージャコード（ＤＭＣ）として設定することが可能である。また、データアップデートチケット-ＤＵＴ（ＰＡＲ）発行手段（DUT Issuer）を、パーティションマネージャが兼ねる構成においては、データアップデートチケット－ＤＵＴ（ＰＡＲ）発行手段（DUT Issuer）のコードは、パーティションマネージャコード（ＰＭＣ）として設定することが可能である。

【０２４９】デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録した［Authentication Type］には、チケットを使用した相互認証として実行すべき認証タイプが記録される。具体的には、後段で詳細に説明するが、デバイス認証、パーティション認証のいずれか、または両方の認証を実行する指定、また公開鍵方式、共通鍵方式のどちらを実行するか、またはいずれの認証でも可能であるかについての情報が記録される。

【０２５０】チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature）、共通鍵方式：ＭＡＣ）を記録する［Integrity Check Value］フィールドには、公開鍵方式であれば、デバイスアップデートチケット発行手段（ＤＵＴ Issuer）の秘密鍵に基づく署名（図１２参照）が生成され格納される。デバイスマネージャ自体がデバイスアップデート登録チケット発行手段（ＤＵ

T Issuer）を兼ねる場合は、デバイスマネージャの秘密鍵を用いて署名が生成される。また、パーティションマネージャ自体がデバイスアップデート登録チケット発行手段（ＤＵＴIssuer）を兼ねる場合は、パーティションマネージャの秘密鍵を用いて署名が生成される。この場合、署名検証処理（図１３参照）の際は、デバイスマネージャまたはパーティションマネージャの公開鍵が用いられる。従って、チケット検証を実行するデバイスは、チケット受領に際し、または前もってデバイスアップデートチケット発行手段（ＤＵＴ issuer）（ｅｘ．デバイスマネージャまたはパーティションマネージャ）の公開鍵（公開鍵証明書）を取得することが必要である。

【０２５１】デバイスアップデートチケット（ＤＵＴ）発行手段（DUT Issuer）の公開鍵証明書（CERT_DUTI）の検証の後、公開鍵証明書（CERT_DUTI）から取り出したデータアップデートチケット（ＤＵＴ）発行手段（DUT Issuer）の公開鍵によりＩＣＶ（Integrity Check Value）の署名検証が可能となる。

【０２５２】データアップデートチケット（ＤＵＴ：Data Update Ticket）を適用して更新されるデータ例を図３３に示す。

【０２５３】図３３に示すように更新対象データには、デバイスマネージャコード、デバイスマネージャコードバージョン、パーティションマネージャコード、パーティションマネージャコードバージョン、各チケット発行手段コード、各チケットのＭＡＣ生成鍵およびバージョン、リボケーションリストなどが含まれる。これら更新対象の各データがデータアップデートチケット（ＤＵＴ：Data Update Ticket）を適用して、ＤＵＴに記録されたルールに従って更新される。更新処理の具体的手順については、後段でフローを用いて説明する。なお、デバイスマネージャコードバージョン、パーティションマネージャコードバージョン他のバージョン情報は、各バージョンの付加されたデータの更新処理の際に併せて更新されることになる。これらのバージョン情報はデータアップデートチケット（ＤＵＴ：Data Update Ticket）に格納される。

【０２５４】［Ｂ．ユーザに対するデバイスの配布、デバイスに対する各種設定、デバイス利用処理の詳細についての説明］次に、上述したパーティション分割されたメモリ領域を持つデバイスの利用に至るまでの処理、さらにデバイスの利用処理の詳細についてフローチャート他の図面を参照しながら説明する。説明の手順は以下の項目に従って行なう。

【０２５５】Ｂ１．デバイス初期登録から利用までの流れ

Ｂ２．デバイス製造エンティテイによる初期登録処理

Ｂ３．デバイスマネージャの管轄処理

Ｂ３．１．デバイスマネージャによるデバイス登録処理

Ｂ３．２．デバイスマネージャ管理下における公開鍵証明書発行処理

Ｂ４．パーティションマネージャの管轄処理

Ｂ４．１．パーティションマネージャ管理下におけるパーティション登録チケット（ＰＲＴ）を利用したパーティション設定登録、削除処理

Ｂ４．２．パーティションマネージャ管理下における公開鍵証明書発行処理

Ｂ４．３．ファイル登録チケット（ＦＲＴ）を利用したファイル生成、消去処理

Ｂ５．サービス許可チケット（ＳＰＴ）を利用したサービス（ファイルアクセス）処理

Ｂ６．データアップデートチケット（ＤＵＴ）を利用したデバイスのデータ更新処理

【０２５６】［Ｂ１．デバイス初期登録から利用までの流れ］ＥＥＰＲＯＭ（フラッシュメモリ）を有するデバイスは、デバイス製造エンティテイ（manufacturer）によって製造され、デバイスマネージャによる初期データの書き込みが実行され、ユーザに提供（ｅｘ．販売、貸与）され利用されることになる。ユーザが様々なサービス主体からデバイスを利用したサービスを受けるためには、デバイスのメモリ部にパーティションマネージャによるパーティションが設定され、設定されたパーティション内にサービス提供用のデータを格納したファイルが設定される必要がある。

【０２５７】また、デバイスに対する様々な処理、すなわちパーティション登録チケット（ＰＲＴ）を利用したパーティションの設定、ファイル登録チケット（ＦＲＴ）を利用したファイル設定、さらにサービス許可チケット（ＳＰＴ）を利用したデータアクセスなどの様々な処理の際に、デバイスとデバイスに対して処理を実行するチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）との間で様々な手続きが実行される。例えば双方が正当な機器、デバイスであることを確認する相互認証処理、あるいは転送データの正当性を保証し確認するための署名生成、検証処理、さらにデータ暗号化、復号処理などである。本発明の構成では、これらの処理に際して公開鍵証明書を用いた構成を提案している。従って、デバイスによるサービスの利用の前にデバイスに対する公開鍵証明書の発行処理、デバイス格納処理を実行する。

【０２５８】例えばデバイスとデバイスに対して処理を実行するチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）との間で公開鍵証明書を用いた相互認証処理が実行され、双方の正当性が確認されたことを条件としてパーティション登録チケット（ＰＲＴ）を利用したパーティションの設定、ファイル登録チケット（ＦＲＴ）を利用したファイル設定、さらにサービス許可チケット（ＳＰＴ）を利用したデータアクセスなどの様々な処理が実行される。また、相互に転送されるデ

ータには必要に応じて電子署名が付加され、検証が実行される。また転送データの暗号化、復号処理も必要に応じて実行されることになる。

【０２５９】図３４は、デバイスの製造から、利用に至るまでの流れを概略的に示した図である。これらの各処理については、フローを参照して後段で詳細に説明するが、全体的な処理の理解のために、図３４に示す各段階について簡単に説明する。

【０２６０】１．まず、デバイスは製造エンティテイ（manufacturer）によって製造される。デバイスの製造時には、各デバイスの識別データ（ＩＤ）としてのデバイスコードが各デバイスに付与される。デバイスには製造段階で、デバイスコード、製造コードなど、様々の製造情報（Manufacture Information Block（図１４参照））が書き込まれデバイスのメモリに格納される。

【０２６１】２．次に、デバイスマネージャはユーザに対するデバイスの提供前に、自己のＩＤ、認証局の公開鍵（ＰＵＢ ＣＡ（ＤＥＶ））など、デバイス管理情報（Device Management Information(図１５参照)）、デバイス鍵（Device Key（図１８参照））などの情報をメモリに格納する。

【０２６２】３．デバイスマネージャによる管理情報が書き込まれたデバイスは、ユーザに提供される。

【０２６３】４．次にユーザは、デバイス対応の公開鍵証明書の取得処理を実行し、取得したデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）をデバイスのデバイス鍵領域（図１８参照）に格納する。

【０２６４】５．デバイスのメモリ部にパーティションを設定し、サービスを提供しようとするサービス主体（パーティションマネージャ）は、パーティションの設定をデバイスマネージャに要求し、許諾を受けるとともにパーティション登録チケット（ＰＲＴ）を受領する。また、デバイスとの通信処理において使用する認証局の公開鍵（ＰＵＢ ＣＡ（ＰＡＲ））を指定する。

【０２６５】６．デバイスは、パーティションマネージャの管理するチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）との間で通信を実行し、パーティション登録チケット（ＰＲＴ）を適用したパーティションの登録処理を行なうとともに、認証局の公開鍵（ＰＵＢ ＣＡ（ＰＡＲ））をパーティション鍵領域（図２３参照）格納する。

【０２６６】７．パーティションの設定されたデバイスは、パーティション対応公開鍵証明書の発行要求をパーティションマネージャに送信し、取得したパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）をパーティション鍵領域（図２３参照）に格納する。

【０２６７】上記５～７のパーティション設定他の処理は、パーティションを設定してサービスを提供しようとするパーティションマネージャ各々について実行され、複数のパーティションがデバイスに登録される。

【０２６８】８．次に、パーティションマネージャは、デバイスに設定したパーティション内に、例えばサービス対応のファイルの設定登録処理をファイル登録チケット（ＦＲＴ）を適用して実行する。

【０２６９】９．１０．設定されたパーティション内にファイルが登録されることにより、例えば電子マネー、定期券などファイル内データによって定義される各種のサービスが実行可能となる。ファイル内のデータ読み取り、データ書き込みなどの処理には、サービス許可チケット（ＳＰＴ）を適用する。すなわち正当なチケット発行手段が発行したサービス許可チケット（ＳＰＴ）を適用した場合に限り、ＳＰＴに記録されたルールに従ってデータの読み取り、書き込みなどが実行される。

【０２７０】また、図には示されていないが、必要に応じてデータアップデートチケット（ＤＵＴ）を使用してデバイスの格納データ中の更新処理対象データ（ｅｘ．デバイスマネージャコード、デバイスマネージャコードバージョン、パーティションマネージャコード、パーティションマネージャコードバージョン、各チケット発行手段コード、各チケットのＭＡＣ生成鍵およびバージョン、リボケーションリストなど）の更新処理が実行される。なお、デバイスマネージャコードバージョン、パーティションマネージャコードバージョン他のバージョン情報は、各バージョンの付加されたデータの更新処理の際に併せて更新されることになる。これらのバージョン情報はデータアップデートチケット（ＤＵＴ：Data Update Ticket）に格納される。

【０２７１】以下、各処理の詳細について、フロー、その他の図を参照しながら説明する。

【０２７２】［Ｂ２．デバイス製造エンティテイによる初期登録処理］まず、デバイス製造エンティテイによる初期登録処理について、図３５を用いて説明する。図３５の左側がデバイス製造エンティテイ(Manufacture)の登録装置の処理、右側がデバイス（図５参照）の処理を示す。なお、デバイス製造エンティテイ(Manufacture)の登録装置は、デバイスに対するデータ読み取り書き込み処理可能な専用のデバイスアクセス機器としてのリーダライタ（図１０参照）として構成される。

【０２７３】まず、ステップＳ１０１において登録装置は、デバイスに対して製造情報ブロック（ＭＩＢ：Manufacture Information Block（図１４参照））の書き込みフラグ（Writable Flag）の読み出しコマンドを送信する。デバイスはコマンドを受信（Ｓ１２１）すると、デバイスのメモリ部の製造情報ブロック（ＭＩＢ）内の書き込み（Writable）フラグを登録装置に送信（Ｓ１２２）する。

【０２７４】製造情報ブロック（ＭＩＢ）内の書き込み（Writable）フラグを受信（Ｓ１０２）した登録装置は、書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されているか否かを判別（Ｓ１

０３）する。書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されていない場合は、以下の製造情報ブロック（ＭＩＢ：Manufacture Information Block）の書き込み処理は実行できず、エラーとして終了する。

【０２７５】書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されている場合は、デバイスの製造情報ブロック（ＭＩＢ：Manufacture Information Block（図１４参照））を生成（Ｓ１０４）してＭＩＢ書き込みコマンドとともに、ＭＩＢデータをデバイスに送信（Ｓ１０５）する。

【０２７６】ＭＩＢ書き込みコマンド、およびＭＩＢデータを受信（Ｓ１２３）したデバイスは、ＭＩＢ書き込みフラグ（Writable Flag）を検証（Ｓ１２４）し、書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されていない場合は、以下の製造情報ブロック（ＭＩＢ：Manufacture Information Block）の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されている場合は、受信したＭＩＢデータをＭＩＢ領域に書き込む（Ｓ１２５）。

【０２７７】ＭＩＢデータ書き込み処理が終了すると書き込み終了通知を登録装置に送信（Ｓ１２６）する。書き込み終了通知を受信（Ｓ１０６）した登録装置は初期登録完了コマンドをデバイスに送信（Ｓ１０７）し、初期登録完了コマンドを受信（Ｓ１２７）したデバイスは製造情報ブロック（ＭＩＢ：Manufacture Information Block）の書き込みフラグ（Writable Flag）を書き込み不可（０ｘ００００）にセット（Ｓ１２８）し、書き込み終了通知を登録装置に送信（Ｓ１２９）する。

【０２７８】書き込み終了通知を受信（Ｓ１０８）した登録装置は、デバイスに対して製造情報ブロック（ＭＩＢ：Manufacture Information Block（図１４参照））の書き込みフラグ（Writable Flag）の読み出しコマンドを送信（Ｓ１０９）する。デバイスはコマンドを受信（Ｓ１３０）すると、デバイスのメモリ部の製造情報ブロック（ＭＩＢ）内の書き込みフラグ（Writable Flag）を登録装置に送信（Ｓ１３１）する。

【０２７９】製造情報ブロック（ＭＩＢ）内の書き込みフラグ（Writable Flag）を受信（Ｓ１１０）した登録装置は、書き込みフラグ（Writable Flag）が書き込み不可（０ｘ００００）に設定されているか否かを判別（Ｓ１１１）する。書き込みフラグ（Writable Flag）が書き込み不可（０ｘ００００）に設定されていない場合は、正常なＭＩＢデータ書き込み処理が終了していないことを示し、エラーとして処理を終了する。書き込みフラグ（Writable Flag）が書き込み不可（０ｘ００００）に設定されている場合は、正常なＭＩＢデータ書き込み処理が終了したものとして処理を終了する。

【０２８０】［Ｂ３．デバイスマネージャの管轄処理］

次に、デバイスマネージャの管轄処理について説明する。ここでは、デバイスの使用開始以前に実行される処理について説明する。デバイスの使用開始以前に実行されるデバイスマネージャの処理としては、デバイスのメモリ部のデバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block）、公開鍵系デバイスキー定義ブロック（ＤＫＤＢ：Device Key Definition Block(PUB)）、共通鍵系デバイスキー定義ブロック（ＤＫＤＢ：Device Key Definition Block(Common)）、デバイス鍵領域（Device Key Area）に対するデータ書き込み処理として実行するデバイス登録処理と、デバイスに対してデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）を発行する処理がある。以下、これらの処理の詳細について説明する。

【０２８１】［Ｂ３．１．デバイスマネージャによるデバイス登録処理］図３６以下のフローを用いて、デバイスマネージャによるデバイスに対するデバイス管理情報他の格納処理を伴う初期登録処理について説明する。図３６以下のフローにおいて、左側がデバイスマネージャ（ＤＭ）の初期登録装置の処理、右側がデバイス（図５参照）の処理を示す。なお、デバイスマネージャ（ＤＭ）の初期登録装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する構成を有する。

【０２８２】まず、ステップＳ２０１において、デバイスの識別子ＩＤｍの読み出し（Read）コマンドをデバイスに出力する。デバイスはコマンドを受信（Ｓ２１１）し、デバイスの識別子ＩＤｍを登録装置に送信（Ｓ２１２）する。

【０２８３】デバイスの識別子ＩＤｍを受信（Ｓ２０２）した登録装置は、ステップＳ２０３において、デバイスに対してデバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block（図１５参照））の書き込みフラグ（Writable Flag）の読み出しコマンドを送信する。デバイスはコマンドを受信（Ｓ２１３）すると、デバイスのメモリ部のデバイス管理情報ブロック（ＤＭＩＢ）内の書き込みフラグ（Writable Flag）を登録装置に送信（Ｓ２１４）する。

【０２８４】デバイス管理情報ブロック（ＤＭＩＢ）内の書き込みフラグ（Writable Flag）を受信（Ｓ２０４）した登録装置は、書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されているか否かを判別（Ｓ２０５）する。書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されていない場合は、以下のデバイス管理情報ブロック（ＤＭＩＢ：Device ManagementInformation Block）の書き込み処理は実行できず、エラーとして終了する。

【０２８５】書き込みフラグ（Writable Flag）が書き

込み可（０ｘｆｆｆｆ）に設定されている場合は、デバイスマネージャコード（DMC）およびDMCバージョンの書き込み（DMC Write）コマンドをデバイスに送信（Ｓ２０６）する。このコードは、コード管理機関（図１～図３参照）によりデバイスマネージャに対して予め割り当てられたデータである。

【０２８６】DMC Writeコマンドを受信（Ｓ２１５）したデバイスは、ＤＭＩＢ書き込みフラグ（Writable Flag）を検証（Ｓ２１６）し、書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されていない場合は、以下のデバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block）の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されている場合は、受信したデバイスマネージャコード（DMC）およびDMCバージョンをＤＭＩＢ領域に書き込む（Ｓ２１７）。

【０２８７】デバイスマネージャコード（DMC）およびDMCバージョンの書き込み処理が終了すると書き込み終了通知を登録装置に送信（Ｓ２１８）する。書き込み終了通知を受信（Ｓ２０７）した登録装置は、次にデバイス総ブロック数（DeviceTotal Block Number）書き込みコマンドをデバイスに送信（Ｓ２０８）する。

【０２８８】デバイス総ブロック数（Device Total Block Number）書き込みコマンドを受信（Ｓ２１９）したデバイスは、ＤＭＩＢ書き込みフラグ（Writable Flag）を検証（Ｓ２２０）し、書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されていない場合は、以下のデバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block）の書き込み処理は実行できず、エラーとして終了する。書き込みフラグ（Writable Flag）が書き込み可（０ｘｆｆｆｆ）に設定されている場合は、受信したデバイス総ブロック数（Device Total Block Number）をＤＭＩＢ領域に書き込む（Ｓ２２１）。さらに、デバイスは、ＤＭＩＢ領域のデバイス空きブロック数情報領域（Free Block Number in Device）にＴＢ－４を書き込む（Ｓ２２２）。ＴＢはデバイス総ブロック数（Device Total Block Number）を意味する。なおＴＢ－４の４ブロックは、製造情報ブロック（ＭＩＢ：Manufacture Information Block）、デバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block）、公開鍵系デバイスキー定義ブロック（ＤＫＤＢ：Device Key Definition Block(PUB)）、共通鍵系デバイスキー定義ブロック（ＤＫＤＢ：Device Key Definition Block(Common)）を示している。

【０２８９】次に、デバイスは、デバイス管理情報ブロック（ＤＭＩＢ）のパーティション数（Partition Number）領域に０を書き込む（Ｓ２２３）。この時点でデバイスにはパーティションは設定されていないからである。さらにＤＭＩＢの空き領域のポインタ（Pointer of Free Area）に０を書き込み（Ｓ２２４）、書き込み処理完了を登録装置に送信（Ｓ２２５）する。

【０２９０】書き込み処理完了通知をデバイスから受信（Ｓ２０９）した登録装置は、次に、デバイス認証に共通鍵を用いるか否かを判定（Ｓ２３１）する。認証処理については、後段で詳細に説明するが、公開鍵認証方式、共通鍵認証方式のいずれかを実行する構成が可能であり、デバイスマネージャは、デバイスに必要な認証方式を設定することが可能となる。デバイスが共通鍵認証を実行するデバイスであれば、デバイスマネージャは共通鍵認証に必要な情報（ｅｘ．認証鍵生成用のマスター鍵他）をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。デバイスマネージャは、デバイスの採用する認証方式に応じて共通鍵認証、公開鍵認証のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【０２９１】図３７に示すように、デバイス認証に共通鍵を用いる場合、ステップＳ２３２～Ｓ２３３、Ｓ２４１～Ｓ２４５を実行し、デバイス認証に共通鍵を用いない場合、これらのステップは省略される。

【０２９２】デバイス認証に共通鍵を用いる場合、ステップＳ２３２において登録装置は、共通鍵認証データ書き込みコマンドとして、MKauth__DEV_A：双方向個別鍵認証用マスター鍵、Kauth__DEV_B：双方向個別鍵認証用共通鍵、IRL_DEV：排除デバイス（Device）のデバイス識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（Device ID)、およびこれらのバージョン情報をデバイスに送信する。

【０２９３】ステップＳ２４１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ２４２において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して受領データをデバイス鍵領域（図１８参照）に書き込む（Ｓ２４３）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ２４４）し、書き込み終了通知を登録装置に送信（Ｓ２４５）する。

【０２９４】書き込み終了通知を受信（Ｓ２３３）した登録装置は、ステップＳ２３４においてデバイス認証に公開鍵を用いるか否かを判定する。図３７に示すように、デバイス認証に公開鍵を用いる場合、ステップＳ２３５～Ｓ２３９、Ｓ２４６～Ｓ２５４を実行し、デバイス認証に公開鍵を用いない場合、これらのステップは省略される。

【０２９５】デバイス認証に公開鍵を用いる場合、ステップＳ２３５において登録装置は、公開鍵認証データ書き込みコマンドとして、PUB_CA(DEV)：デバイスマネージャ対応公開鍵を発行する認証局ＣＡ（ＤＥＶ）の公開鍵、PARAM_DEV：デバイス（Device）の公開鍵パラメー

タ、CRL_DEV :排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate)、およびこれらのバージョン情報をデバイスに送信する。

【０２９６】ステップＳ２４６でデバイスは、上述の書き込みコマンドを受信し、ステップＳ２４７において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して受領データをデバイス鍵領域（図１８参照）に書き込む（Ｓ２４８）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ２４９）し、書き込み終了通知を登録装置に送信（Ｓ２５０）する。

【０２９７】書き込み終了通知を受信（Ｓ２３６）した登録装置は、公開鍵と秘密鍵の鍵ペア生成コマンドをデバイスに送信（Ｓ２３７）する。なお、この実施例では、鍵ペアの生成はデバイスが実行する構成としているが、例えば登録装置が実行してデバイスに提供する構成としてもよい。

【０２９８】鍵ペア生成コマンドを受信（Ｓ２５１）したデバイスは、デバイス内の暗号処理部（図５参照）において公開鍵（ＰＵＢ ＤＥＶ）と秘密鍵（ＰＲＩ ＤＥＶ）のペアを生成し、生成した鍵をデバイス鍵領域（図１８参照）に書き込む（Ｓ２５２）。なお、公開鍵（ＰＵＢ ＤＥＶ）については、デバイス鍵領域のＣＥＲＴ・ＤＥＶ領域に一時格納し、その後、公開鍵（ＰＵＢ ＤＥＶ）を格納した公開鍵証明書を受領した時点で公開鍵証明書（ＣＥＲＴ）に置き換えられる。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ２５３）し、生成格納した公開鍵を登録装置に送信（Ｓ２５４）する。

【０２９９】登録装置は、デバイスから公開鍵（ＰＵＢ ＤＥＶ）を受信し、先にデバイスから受信したデバイスの識別子ＩＤｍとともに、デバイスマネージャ内のデータベース（ＤＢ（ＤＥＶ）（図７参照））に保存する。

【０３００】次に、デバイスマネージャの登録装置は、パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）の検証処理に共通鍵を用いるか否かを判定（Ｓ２６１）する。チケット検証には、後段で詳細に説明するがＭＡＣ値検証等による共通鍵方式と、前述の図１２、図１３を用いて説明した秘密鍵による署名生成、公開鍵による署名検証を行なう公開鍵方式のいずれかを適用することが可能であり、デバイスマネージャは、デバイスの採用する検証処理方式を設定することができる。デバイスマネージャは、デバイスの採用するＰＲＴチケット検証方式に応じて共通鍵、公開鍵のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【０３０１】デバイスマネージャは、デバイスが共通鍵認証を実行するデバイスであれば、デバイスマネージャは共通鍵方式のＰＲＴ検証に必要な情報（ｅｘ．ＰＲＴ検証共通鍵）をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【０３０２】図３８に示すように、ＰＲＴ検証に共通鍵方式を用いる場合、ステップＳ２６２～２６３、Ｓ２７１～Ｓ２７５を実行し、ＰＲＴ検証に共通鍵を用いない場合、これらのステップは省略される。

【０３０３】ＰＲＴ検証に共通鍵を用いる場合、ステップＳ２６２において登録装置は、ＰＲＴ検証共通鍵書き込みコマンドとして、Kprt :パーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵、およびバージョン情報をデバイスに送信する。

【０３０４】ステップＳ２７１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ２７２において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して受領データをデバイス鍵領域（図１８参照）に書き込む（Ｓ２７３）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ２７４）し、書き込み終了通知を登録装置に送信（Ｓ２７５）する。

【０３０５】書き込み終了通知を受信（Ｓ２６３）した登録装置は、ステップＳ２６４においてＰＲＴ検証に公開鍵を用いるか否かを判定する。図３８に示すように、ＰＲＴ検証に公開鍵を用いる場合、ステップＳ２６５～Ｓ２６６、Ｓ２７６～Ｓ２８２を実行し、ＰＲＴ検証に公開鍵を用いない場合、これらのステップは省略される。

【０３０６】ＰＲＴ検証に公開鍵を用いる場合、ステップＳ２６５において登録装置は、ＰＲＴ検証データ書き込みコマンドとして、PRTIC（PRT Issuer Category）：パーティション登録チケット（ＰＲＴ）発行者カテゴリ、PUB_CA(DEV) :デバイスマネージャ対応公開鍵を発行する認証局ＣＡ（ＤＥＶ）の公開鍵、PARAM_DEV :デバイス（Device）の公開鍵パラメータ、CRL_DEV :排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate)、およびこれらのバージョン情報をデバイスに送信する。

【０３０７】ステップＳ２７６でデバイスは、上述の書き込みコマンドを受信し、ステップＳ２７７において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して、ステップＳ２７８において、受領データ中のPRTIC（PRTIssuer Category）：パーティション登録チケット（ＰＲＴ）発行者カテゴリを公開鍵系デバイス鍵定義ブロック（ＤＫＤＢ：Device Key Definition block（PUB）（図１６参照））に書き込みバージョン情報を同ブロックのバージョン領域に書き込む。

【０３０８】次にデバイスは、ステップＳ２７９におい

て、PUB_CA(DEV) :デバイスマネージャ対応公開鍵を発行する認証局ＣＡ（ＤＥＶ）の公開鍵データが書き込み済みか否かを判定し、書き込まれていない場合にステップＳ２８０において、PUB_CA(DEV)、PARAM_DEV、CRL_DEVをデバイス鍵領域（図１８参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ２８１）し、書き込み終了通知を登録装置に送信（Ｓ２８２）する。

【０３０９】書き込み終了通知を受信（Ｓ２６６）した登録装置は、次に、ステップＳ２９１において、共通鍵データの更新をサポートするデバイスであるか否かを判定する。デバイスに格納されたデータ中、そのいくつかは更新対象データとして前述したデータアップデートチケット（ＤＵＴ：Data Update Ticket）（図３２参照）を用いて更新が可能である。更新対象となるデータは、先に図３３を用いて説明した通りである。このデータアップデートチケット（ＤＵＴ：Data Update Ticket）を用いた更新処理においても共通鍵方式、または公開鍵方式のいずれかの方式が可能であり、デバイスマネージャはデバイスに応じていずれかの方式または両方式を実行可能なデータをデバイスに設定する。

【０３１０】デバイスマネージャは、デバイスが共通鍵方式によるデータ更新を実行するデバイスであれば、共通鍵方式のデータ更新処理に必要な情報（ｅｘ．データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵他）をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【０３１１】図３９に示すように、データアップデートチケット（ＤＵＴ：Data Update Ticket）を用いたデータ更新処理に共通鍵方式を用いる場合、ステップＳ２９２～Ｓ２９３、Ｓ３０１～Ｓ３０５を実行し、データ更新に共通鍵方式を用いない場合、これらのステップは省略される。

【０３１２】データ更新に共通鍵を用いる場合、ステップＳ２９２において登録装置は、データアップデートチケット（ＤＵＴ：Data Update Ticket）検証共通鍵書き込みコマンドとして、Kdut_DEV1 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵、 Kdut_DEV2 :データ更新用暗号鍵、Kdut_DEV3 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵、Kdut_DEV4 :データ更新用暗号鍵およびこれらのバージョン情報をデバイスに送信する。

【０３１３】ステップＳ３０１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ３０２において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して受領データをデバイス鍵領域（図１８参照）に書き込む（Ｓ３０３）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ３０４）し、書き込み終了通知を登録装置に送信（Ｓ３０５）する。

【０３１４】書き込み終了通知を受信（Ｓ２９３）した登録装置は、ステップＳ２９４において、デバイスが公開鍵方式を用いたデータアップデートチケット（ＤＵＴ：Data Update Ticket）を使用したデータ更新処理をサポートするか否かを判定する。図３９に示すように、公開鍵方式をサポートする場合、ステップＳ２９５～Ｓ２９６、Ｓ３０６～Ｓ３１０を実行し、公開鍵方式をサポートしない場合、これらのステップは省略される。

【０３１５】公開鍵方式をサポートする場合、ステップＳ２９５において登録装置は、データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者コード書き込みコマンドとして、DUTIC_DEV (DUT Issuer Category) :データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者カテゴリ、およびバージョン情報をデバイスに送信する。

【０３１６】ステップＳ３０６でデバイスは、上述の書き込みコマンドを受信し、ステップＳ３０７において、ＤＭＩＢの書き込みフラグ（Writable Flag）が書き込み可であることを確認して、ステップＳ３０８において、受領データを公開鍵系デバイス鍵定義ブロック（ＤＫＤＢ（ＰＵＢ）：Device Key Definition Block(PUB)）に書き込む（Ｓ３０８）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ３０９）し、書き込み終了通知を登録装置に送信（Ｓ３１０）する。

【０３１７】書き込み終了通知を受信（Ｓ２９６）した登録装置は、次にステップＳ３２１において、デバイスマネージャ（DM）初期登録完了コマンドをデバイスに対して送信する。コマンドを受領（Ｓ３３１）したデバイスは、ステップＳ３３２において、相互認証、パーティション登録チケット（ＰＲＴ）の検証、さらにデータアップデートチケット（ＤＵＴ）の検証、それぞれについて少なくとも公開鍵方式、共通鍵方式のいずれかの処理が実行可能なデータが設定済みであるか否かを判定する。これらのデータに不足がある場合は、いずれかの処理が実行できないことになり、デバイスマネージャによる初期登録はエラーと判定され処理を終了する。

【０３１８】ステップＳ３３２において、相互認証、パーティション登録チケット（ＰＲＴ）の検証、さらにデータアップデートチケット（ＤＵＴ）の検証、それぞれについて少なくとも公開鍵方式、共通鍵方式のいずれかの処理が実行可能なデータが設定済みであると判定した場合は、ステップＳ３３３においてデバイスは、デバイス管理情報ブロック（ＤＭＩＢ：Device Management Information Block）の書き込み（Writable）フラグを書き込み不可（０ｘ０００００）にセットし、書き込み終了通知を登録装置に送信(Ｓ３３４)する。

【０３１９】書き込み終了通知を受信（Ｓ３２２）した登録装置は、デバイスに対してデバイス管理情報ブロッ

ク（ＤＭＩＢ：Device Management Information Block）（図１５参照））の書き込みフラグ（Writable Flag）の読み出しコマンドを送信（Ｓ３２３）する。デバイスはコマンドを受信（Ｓ３３５）すると、デバイスのメモリ部のデバイス管理情報ブロック（ＤＭＩＢ）内の書き込みフラグ（Writable Flag）を登録装置に送信（Ｓ３３６）する。

【０３２０】デバイス管理情報ブロック（ＤＭＩＢ）内の書き込みフラグ（Writable Flag）を受信（Ｓ３２４）した登録装置は、書き込みフラグ（Writable Flag）が書き込み不可（０ｘ０００００）に設定されているか否かを判別する。書き込みフラグ（Writable Flag）が書き込み不可（０ｘ００００）に設定されていない場合は、正常なＤＭＩＢデータ書き込み処理が終了していないことを示し、エラーとして処理を終了する。書き込みフラグ（Writable Flag）が書き込み不可（０ｘ０００００）に設定されている場合は、正常なＤＭＩＢデータ書き込み処理が終了したものとして処理を終了する。

【０３２１】デバイス製造エンティテイ(Manufacture)の登録装置による初期登録（図３５の処理フロー）および、デバイスマネージャによる初期登録処理（図３６～図４０の処理フロー）が完了した状態のデバイスのメモリ内格納データ構成例を図４１に示す。図４１は、図６、図１４乃至図１８を用いて説明した製造情報ブロック（Manufacture Information Block）、デバイス管理情報ブロック（Device Management Information Block）、公開鍵系デバイス鍵定義（Device Key Definition Block(PUB)）、共通鍵系デバイス鍵定義ブロック（Device Key DefinitionBlock(Common)）、デバイス鍵領域（Device Key Area）を示すものである。この時点では、メモリにパーティションは形成されていない。

【０３２２】製造情報ブロック（Manufacture Information Block）には、図１４を用いて説明したように、デバイスの固有情報としてのデバイスコード他が書き込まれる。この製造情報ブロック（Manufacture Information Block）に書き込まれた情報、あるいは書き込まれた情報の一部、または書き込まれた情報に基づいて取得される演算データがデバイスの識別子（ＩＤｍ）に相当する。

【０３２３】なお、図に示すデバイス鍵領域（Device Key Area）には Kauth_DEV_B：双方向個別鍵認証用共通鍵、 MKauth_DEV_A：双方向個別鍵認証用マスター鍵が格納されているが、これらの鍵は、デバイスが共通鍵認証処理を行なう要請が無い場合は格納しない構成としてもよく、また、Kprt：パーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵についても、デバイスが共通鍵によるチケット検証処理を実行しない構成の場合には格納しない構成としてもよい。

【０３２４】また、IRL_DEV：排除デバイス（Device）のデバイス識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（Device ID））、CRL_DEV：排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate)についても、デバイス発行時点でリボーク（排除）されたデバイスが存在しない場合、あるいは他のソースを使用してリボケーションリストを取得する構成とする場合には、リボケーションリストを格納しない構成としてもよい。

【０３２５】［Ｂ３．２．デバイスマネージャ管理下における公開鍵証明書発行処理］次に図４２以下を用いて、デバイスマネージャによるデバイス対応公開鍵証明書の発行処理について説明する。デバイスには、デバイス全体の認証、デバイスを単位とした処理に適用可能なデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）と、デバイス内の特定のパーティションに対する処理の際の認証その他検証処理等に適用可能なパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）が格納され得る。パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）は、デバイスに設定されたパーティション毎に設定格納可能である。

【０３２６】デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）は、デバイスマネージャの管轄するメモリ領域であるデバイス鍵領域（Device Key Area）（図１８参照）に格納され、パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）は、各パーティションマネージャの管轄するメモリ領域であるパーティション鍵領域（Partition Key Area）（図２３参照）に格納される。

【０３２７】デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）は、デバイスマネージャの管轄する登録局を介して認証局（ＣＡ ｆｏｒ ＤＭ）（図２、図３参照）の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、デバイスマネージャの管轄登録局が発行した公開鍵証明書（ＣＥＲＴ ＤＥＶ）についての管理（データベース２２２（図７参照））を実行する。

【０３２８】またパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）は、パーティションマネージャの管轄する登録局を介して認証局（ＣＡ ｆｏｒ ＰＭ）（図２、図３参照）の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、パーティションマネージャの管轄登録局が発行した公開鍵証明書（ＣＥＲＴ ＰＡＲ）についての管理（データベース３３２（図９参照））を実行する。

【０３２９】図４２および図４３に従って、デバイスマネージャの管轄登録局によるデバイスに対するデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行処理の手順を説明する。なお、デバイスマネージャの登録局（ＲＡ）構成のみを取り出した発行装置（ＤＭ）、認証局（ＣＡ）、ユーザデバイスとの関係を図４４に示した。図４４に示すように制御手段２２１は暗号処理手段を有する。なお暗号処理は暗号処理に関するプログラムを制

御部（ＣＰＵ（図８の２１１１）の制御下において実行することによって行われる。

【０３３０】図４２，図４３において、左側がデバイスマネージャの管轄登録局のＣＥＲＴ（公開鍵証明書）発行装置、具体的には、図７に示すデバイスマネージャの構成図における制御手段２２１の処理、右側がデバイスの処理である。

【０３３１】まずステップＳ３５１において、ＣＥＲＴ発行装置は、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行対象となるデバイスのユーザ情報を取得し、証明書発行の許可（判定）を行ない発行対象となるデバイスとの通信路を確保する。デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行対象となるデバイスのユーザ情報は、例えばデバイスの初期登録時に生成したデータから取得可能である。また、別途、別経路にてユーザの名前や住所、電話番号、ｅ－ｍａｉｌアドレスなどを取得するようにしてもよい。なお、ユーザ情報はデバイスとの通信路設定後、デバイスから取得してもよい。通信路は、有線、無線を問わずデータ送受信可能な通信路として確保されればよい。

【０３３２】次にＣＥＲＴ発行装置は、ステップＳ３５２において、乱数を含む認証データ生成コマンドをデバイスに対して送信する。認証データ生成コマンドを受信（Ｓ３６１）したデバイスは、受信乱数Ｒと、デバイス識別子（ＩＤｍ）の結合データにデバイス秘密鍵（ＰＲＩ ＤＥＶ）を適用してデジタル署名（Ｓ）の生成処理（図１２参照）を実行（Ｓ３６２）する。デバイスは、デバイスの識別データ（ＩＤｍ）と署名（Ｓ）をＣＥＲＴ発行装置に送信する。

【０３３３】デバイスから識別データ（ＩＤｍ）と署名（Ｓ）を受信（Ｓ３５３）したＣＥＲＥＴ発行装置は、受信したデバイス識別データ（ＩＤｍ）を検索キーとしてデータベースＤＢ（ＤＥＶ）２２２から格納済みのデバイス公開鍵（ＰＵＢ ＤＥＶ）を取得する。さらに、取得したデバイス公開鍵（ＰＵＢ ＤＥＶ）を適用して署名（Ｓ）の検証処理（図１３参照）を実行（Ｓ３５５）する。検証に成功しなかった場合は、デバイスからの送信データは不正なデータであると判定し処理は終了する。

【０３３４】検証に成功した場合は、認証局（ＣＡ ｆｏｒ ＤＭ）６１０に対してデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行処理を依頼（Ｓ３５７）する。デバイスマネージャは認証局６１０の発行したデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）を受信（Ｓ３５８）してデバイスに送信（Ｓ３５９）する。

【０３３５】デバイスマネージャ（登録局）からデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）を受信したデバイスは、予めデバイス鍵領域に格納済みの認証局の公開鍵（ＰＵＢ ＣＡ（ＤＥＶ））を用いて受信したデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の署名検証を実行する。すなわち公開鍵証明書には認証局の秘密鍵で実行された署名があり（図１１参照）、この署名検証（Ｓ３６６）を行なう。

【０３３６】署名検証に失敗した場合は、正当な公開鍵証明書でないと判定し、エラー通知をＣＥＲＴ発行装置に対して実行（Ｓ３８５）する。

【０３３７】署名検証に成功した場合は、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）に格納されたデバイス公開鍵（ＰＵＢ ＤＥＶ）と自デバイスに保管されたデバイス公開鍵（ＰＵＢ ＤＥＶ）の比較を実行（Ｓ３８２）し、一致しない場合はエラー通知を実行し、一致した場合は、受信したデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）をデバイス鍵領域（図１８参照）に格納（Ｓ３８３）する。なお、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行以前は、この領域に自デバイスで生成した公開鍵（ＰＵＢ ＤＥＶ）を格納し、正当なデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）が発行された時点で、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）により上書きする処理として格納する。

【０３３８】デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の格納が終了すると格納処理終了通知をＣＥＲＴ発行装置に送信（Ｓ３８４）する。ＣＥＲＴ発行装置は、格納処理終了通知を受信（Ｓ３７１）し、格納成功を確認（Ｓ３７２）して処理を終了する。格納成功の確認が得られない場合はエラーとして処理が終了する。

【０３３９】図４５にデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行処理において、デバイスマネージャ２００、デバイス１００、認証局（ＣＡ）６１０各エンティティ間のデータ送受信処理について説明した図を示す。

【０３４０】図４５中のＮｏ．１～１４の順に処理が実行される。なお、処理Ｎｏ．１のデバイスマネージャ２００によるデバイス１００からのデバイス識別子（ＩＤｍ）、デバイス公開鍵（ＰＵＢ ＤＥＶ）の取得処理、および処理Ｎｏ．２のデバイス識別子（ＩＤｍ）の登録処理はデバイスマネージャによる初期登録において実行される処理である。

【０３４１】デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の発行手続きは、処理Ｎｏ．３からであり、３．デバイスマネージャによるデバイスからの顧客情報取得、４．顧客情報の登録（登録済みである場合は不要）、５．デバイスからのデバイス識別子（ＩＤｍ）取得、６．取得したデバイス識別子（ＩＤｍ）にもとついてデータベース検索を実行して対応の公開鍵（ＰＵＢ ＤＥＶ）を取得、７．デバイスとデバイスマネージャ間における認証処理、この処理は図４２、図４３の処理においては省略されていたが、図４２、図４３においてはデバイスからのデバイス識別子（ＩＤｍ）取得の際に署名検証を実行しており、通信相手の送信データの確認により、認証を省略した。図４２、図４３における署名検

証、図４５の認証の少なくともいずれか、またはいずれをも実行することが望ましい。なお、認証処理の詳細については、後段のＢ．４．パーティションマネージャの管轄処理の項目で説明する。

【０３４２】８．デバイスマネージャから認証局に対するデバイス対応公開鍵証明書の発行要求、９．デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の生成、１０．認証局における生成公開鍵証明書のデータ登録、１１．認証局（ＣＡ）６１０からデバイスマネージャ２００に対する公開鍵証明書の配布、１２．デバイスマネージャのデータベース更新（公開鍵証明書発行情報登録）、１３．デバイスマネージャからデバイスに対するデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の送信、１４．デバイスにおけるデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）の保存、保存は、前述したようにデバイス鍵領域に書き込み保存される。

【０３４３】以上の処理により、デバイスは、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）を取得し、メモリ部に格納する。このデバイス対応公開鍵証明書（ＣＥＲＴＤＥＶ）をメモリのデバイス鍵格納領域に格納した後のメモリの各ブロックのデータ格納構成を図４６に示す。図４６は、先に図６、図１４乃至図１８を用いて説明した製造情報ブロック（Manufacture Information Block）、デバイス管理情報ブロック（Device Management Information Block）、公開鍵系デバイス鍵定義（Device Key Definition Block(PUB)）、共通鍵系デバイス鍵定義ブロック（Device Key Definition Block(Common)）、デバイス鍵領域（Device Key Area）を示すものである。この時点では、メモリにパーティションは形成されていない。

【０３４４】図４６に示すデバイス鍵領域（Device Key Area）にはデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）が格納される。デバイス対応公開鍵証明書（ＣＥＲＴＤＥＶ）の発行前には、この領域には、デバイスが生成した公開鍵（ＰＵＢ ＤＥＶ）が格納され、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）を受信すると、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）によって上書き処理がなされる。なお、この上書き処理によりポインタ、サイズ、管理データがある場合は必要な変更処理を実行する。

【０３４５】［Ｂ４．パーティションマネージャの管轄処理］次に、パーティションマネージャの管轄処理について説明する。ここでは、デバイスの使用開始以前に実行される処理について説明する。デバイスの使用開始以前に実行されるパーティションマネージャの処理としては、デバイスのメモリ部にパーティションを設定する処理と、デバイスに対してパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）を発行する処理がある。以下、これらの処理の詳細について説明する。パーティションを設定する処理には、デバイスとパーティションマネージャ間における相互認証処理（デバイス認証またはパーティション認証）、パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）の正当性検証処理が含まれる。なお、パーティションの削除処理についても基本的にパーティション作成と同様の手続きに従って実行可能であるので併せて説明する。

【０３４６】［Ｂ４．１．パーティションマネージャ管理下におけるパーティション登録チケット（ＰＲＴ）を利用したパーティション設定登録、削除処理］まず、パーティション登録チケット（ＰＲＴ）（図２６参照）を利用したパーティション設定登録、削除処理について説明する。図４７以下のフロー他の図面を参照して説明する。なお、上述のように、パーティション設定処理には、デバイスとパーティションマネージャ間における相互認証処理（デバイス認証またはパーティション認証）、パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）の正当性検証処理が含まれ、これらの処理についても説明する。

【０３４７】図４７に示すパーティション設定登録、削除処理フローについて説明する。図４７において、左側がパーティションマネージャのパーティション作成・削除装置、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャのパーティション作成・削除装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する。まず、図４７を用いて、パーティション作成、削除処理の概要を説明し、その後、当処理に含まれる各処理の詳細を図４８以下のフローを用いて順次説明する。

【０３４８】まず、図４７のステップＳ４０１とＳ４１０において、パーティション作成・削除装置とデバイス間での相互認証処理が実行される。データ送受信を実行する２つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう構成が１つの好ましいデータ転送方式である。

【０３４９】本発明のシステムにおける相互認証処理では、デバイス認証またはパーティション認証のいずれかが実行される。また、それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。これらについては後述する。

【０３５０】なお、相互認証処理として実行すべき処理は、適用するパーティション登録チケット（ＰＲＴ）（図２６参照）の

＊ Authentication Flag ：チケット（Ticket）の利用処

理においてデバイス（Device）との相互認証が必要か否かを示すフラグ
* Authentication Type :デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））
によって決定される。

【０３５１】認証処理に失敗した場合（Ｓ４０２，Ｓ４１１でＮｏ）は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【０３５２】認証処理に成功すると、パーティション作成・削除装置は、デバイスに対してパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）を送信する。パーティション登録チケット（ＰＲＴ）は、パーティションマネージャの要求により、デバイスマネージャの管理するパーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）によりパーティションマネージャに対して発行されるチケットである。パーティション登録チケット（ＰＲＴ）は、デバイスに対するアクセス制御チケットであり、先に説明した図２６のデータフォーマット構成を持つチケットである。

【０３５３】なお、パーティション登録チケット（ＰＲＴ）を、チケットユーザに対して送信する際には、公開鍵方式の場合、パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の公開鍵証明書（CERT_PRTI）も一緒に送信する。ＰＲＴ発行手段の公開鍵証明書（CERT_PRTI）の属性（Attribute）は、パーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ User）の識別子(PRTIC)と一致する。

【０３５４】パーティション登録チケット（ＰＲＴ）を受信（Ｓ４１２）したデバイスは、受信したチケット（ＰＲＴ）の正当性と利用者チェック処理を実行（Ｓ４１３）する。チケットの正当性の検証処理は、共通鍵方式によるＭＡＣ検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器（チケット利用者）の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子（図２６参照）との一致等を検証する処理として実行される。これらの処理の詳細については後述する。

【０３５５】デバイスにおいて、受信チケット（ＰＲＴ）の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合（Ｓ４１４でＮｏ）は、パーティション登録チケット（ＰＲＴ）受理エラーをパーティション作成・削除装置に通知（Ｓ４１８）する。チケットおよび利用者の正当なことが確認できた場合（Ｓ４１４でＹｅｓ）は、受信したパーティション登録チケット（ＰＲＴ）に記述されたルールに従いデバイス内のメモリ部におけるパーティションの生成、または削除処理を実行する。この処理の詳細についても別フローを用いて後段で詳述する。

【０３５６】パーティション登録チケット（ＰＲＴ）の記述に従って、パーティションの生成または削除処理に成功（Ｓ４１６でＹｅｓ）すると、ＰＲＴ受理成功をパーティション作成・削除装置に通知（Ｓ４１７）する。一方、パーティションの生成または削除処理に失敗（Ｓ４１６でＮｏ）した場合は、ＰＲＴ受理エラーをパーティション作成・削除装置に通知（Ｓ４１８）する。

【０３５７】パーティション作成・削除装置は、ＰＲＴ受理結果を受信（Ｓ４０４）し、ＰＲＴ処理結果を判定し、ＰＲＴ受理結果がエラーである場合（Ｓ４０５でＮｏ）は、エラーとして処理を終了し、ＰＲＴ受理結果が成功（Ｓ４０５でＹｅｓ）であり、処理がパーティション生成である場合はパーティション初期データの書き込み処理（Ｓ４０６，Ｓ４１９）を実行する。初期データの書き込み処理については、後段で別フローを用いて詳述する。パーティション初期データの書き込み処理が終了した場合、およびＰＲＴ受理結果が成功（Ｓ４０５でＹｅｓ）であり、処理がパーティション削除である場合はセッションクリアコマンドの送受信（Ｓ４０７，Ｓ４２０）を実行し、デバイス側に生成した認証テーブルを破棄（Ｓ４２１）し、処理を終了する。認証テーブルは、ステップＳ４０１，Ｓ４１０の相互認証処理において生成されるテーブルであり、その詳細は後述する。

【０３５８】このようにパーティション登録チケット（ＰＲＴ）を利用して、デバイス内に新たなパーティションの生成、または生成済みのパーティションの削除が実行される。以下、当処理に含まれる相互認証処理（Ｓ４０１，Ｓ４１０）、チケットの正当性と利用者チェック（Ｓ４１３）、パーティションの生成、削除処理（Ｓ４１５）、パーティション初期データ書き込み処理（Ｓ４０６，Ｓ４１９）の各処理について詳述する。

【０３５９】（相互認証処理）図４７のステップＳ４０１，Ｓ４１０において実行される相互認証処理について説明する。なお、以下に説明する相互認証処理は、他のチケット、すなわちファイル登録チケット（ＦＲＴ：File Registration Ticket）、サービス許可チケット（ＳＰＴ：Service Permission Ticket）、データアップデートチケット（ＤＵＴ：Data Update Ticket）を使用したデバイスアクセス処理においても適宜必要に応じて行われる処理である。

【０３６０】相互認証処理の処理フローを図４８に示す。図４８において、左側がパーティションマネージャの認証装置、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャの認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のリーダライタに相当する構成を有する。

【０３６１】図４８のステップＳ４３１において、認証装置は、パーティション登録チケット（ＰＲＴ）の利用に必要な認証方式をチケットから読み出して決定する。なお、認証態様は、チケット記載の認証方式に限らず、例えばアクセス機器（ｅｘ．リーダライタ）からの指定された方式に従ってデバイス認証、パーティション認証を決定してもよい。

【０３６２】決定した認証方式をＡ（１）～Ａ（ｎ）とする。パーティション登録チケット（ＰＲＴ）を適用したパーティションの設定登録、あるいは削除処理においては様々な認証処理態様が設定される。例えばパーティションの設定登録についてはデバイスを対象とするデバイス認証を必要とし、パーティション削除の場合にはデバイス認証と削除対象となるパーティション認証の双方を必要とするなどである。このように処理に応じていずれか一方の認証、または両者の認証を必要とする設定をパーティション登録チケット（ＰＲＴ）に記述することができる。ＰＲＴ利用処理において必要とする認証方式については、パーティション登録チケット（ＰＲＴ）の［Authentication Type］に記述され、認証装置は、パーティション登録チケット（ＰＲＴ）の利用に必要な認証方式をチケットから読み出して決定し、認証処理手順をＡ（ｉ）：Ａ（１）～Ａ（ｎ）として定義する。

【０３６３】ステップＳ４３２において、最初の認証処理方式Ａ（１）を読み出し、Ａ（１）の認証方式がデバイス認証、パーティション認証であるかについて判別（Ｓ４３３）し、デバイス認証であればデバイス認証を実行（Ｓ４３４，Ｓ４４１）し、パーティション認証であればパーティション認証を実行（Ｓ４３５，Ｓ４４２）する。デバイスとの認証処理の結果、認証が成功しない場合は、エラーとして処理を終了する。認証が成功した場合は、ステップＳ４３７においてｉをインクリメントしてステップＳ４３３に戻り、次の認証方式を判別して、方式に従って認証を実行する。これらの処理をＡ（１）～Ａ（ｎ）まで実行して、全ての認証が成功した場合に次のステップに進む。

【０３６４】デバイス認証処理について図４９のフローに従って説明する。図４９において、左側がパーティションマネージャのデバイス認証装置、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャのデバイス認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する構成を有する。

【０３６５】デバイス認証装置は、ステップＳ４５１において、パーティション登録チケット（ＰＲＴ）に基づいてデバイス認証処理に公開鍵を用いた公開鍵認証方式を適用するか否かを判定する。デバイス認証は、公開鍵方式または共通鍵方式のいずれかにおいて実行され、チケットにその実行方式についての指定が記述されている。

【０３６６】共通鍵方式の指定がある場合は、図４９のステップＳ４５２～Ｓ４５５、Ｓ４６１～Ｓ４６５の処理は実行されず、ステップＳ４５６に進む。公開鍵方式の指定である場合は、デバイス認証装置は、ステップＳ４５２において公開鍵デバイス認証開始コマンドをデバイスに送信する。デバイスは、コマンドを受信（Ｓ４６１）すると、デバイスのメモリ部の公開鍵系デバイス鍵定義ブロック（図１６参照）を参照して、デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）が格納されているか否かを検証（Ｓ４６２）する。デバイス対応公開鍵証明書（ＣＥＲＴＤＥＶ）が格納されていない場合は、公開鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【０３６７】デバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）が格納されていることが確認されると、ステップＳ４５３、Ｓ４６３において、デバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の発行した公開鍵証明書を用いた相互認証および鍵共有処理が実行される。

【０３６８】公開鍵方式による相互認証および鍵共有処理について、図５０を用いて説明する。図５０は、公開鍵暗号方式である１６０ビット長の楕円曲線暗号（ＥＣＣ）を用いた相互認証シーケンスを示している。図５０では、公開鍵暗号方式としてＥＣＣを用いているが、公開鍵暗号方式の他方式を適用することも可能である。また、鍵サイズも１６０ビットでなくてもよい。図５０において、まずＢが、６４ビットの乱数Ｒｂを生成し、Ａに送信する。これを受信したＡは、新たに６４ビットの乱数Ｒａおよび標数ｐより小さい乱数Ａｋを生成する。そして、ベースポイントＧをＡｋ倍した点Ａｖ＝Ａｋ×Ｇを求め、Ｒａ、Ｒｂ、Ａｖ（X座標とY座標）に対する電子署名Ａ．Ｓｉｇを生成し、Ａの公開鍵証明書とともにＢに返送する。ここで、ＲａおよびＲｂはそれぞれ６４ビット、ＡｖのＸ座標とＹ座標がそれぞれ１６０ビットであるので、合計４４８ビットに対する電子署名を生成する。

【０３６９】公開鍵証明書を利用する際には、利用者は自己が保持する公開鍵証明書発行局（ＣＡ）の公開鍵を用い、当該公開鍵証明書の電子署名を検証し、電子署名の検証に成功した後に公開鍵証明書から公開鍵を取り出し、当該公開鍵を利用する。従って、公開鍵証明書を利用する全ての利用者は、共通の公開鍵証明書発行局（ＣＡ）の公開鍵を保持している必要がある。なお、電子署名の検証方法については、図１３で説明したのでその詳細は省略する。

【０３７０】Ａの公開鍵証明書、Ｒａ、Ｒｂ、Ａｖ、電子署名Ａ．Ｓｉｇを受信したＢは、Ａが送信してきたＲｂが、Ｂが生成したものと一致するか検証する。その結果、一致していた場合には、Ａの公開鍵証明書内の電子

署名を認証局の公開鍵で検証し、Aの公開鍵を取り出す。そして、取り出したAの公開鍵を用い電子署名A.Sigを検証する。電子署名の検証に成功した後、BはAを正当なものとして認証する。

【0371】次に、Bは、標数pより小さい乱数Bkを生成する。そして、ベースポイントGをBk倍した点Bv＝Bk×Gを求め、Rb、Ra、Bv（X座標とY座標）に対する電子署名B.Sigを生成し、Bの公開鍵証明書とともにAに返送する。

【0372】Bの公開鍵証明書、Rb、Ra、Bv、電子署名B.Sigを受信したAは、Bが送信してきたRaが、Aが生成したものと一致するか検証する。その結果、一致していた場合には、Bの公開鍵証明書内の電子署名を認証局の公開鍵で検証し、Bの公開鍵を取り出す。そして、取り出したBの公開鍵を用い電子署名B.Sigを検証する。電子署名の検証に成功した後、AはBを正当なものとして認証する。

【0373】両者が認証に成功した場合には、BはBk×Av（Bkは乱数だが、Avは楕円曲線上の点であるため、楕円曲線上の点のスカラー倍計算が必要）を計算し、AはAk×Bvを計算し、これら点のX座標の下位64ビットをセッション鍵として以降の通信に使用する（共通鍵暗号を64ビット鍵長の共通鍵暗号とした場合）。もちろん、Y座標からセッション鍵を生成してもよいし、下位64ビットでなくてもよい。なお、相互認証後の秘密通信においては、送信データはセッション鍵で暗号化されるだけでなく、電子署名も付されることがある。

【0374】電子署名の検証や受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中断する。

【0375】このような相互認証処理において、生成したセッション鍵を用いて、送信データを暗号化して、相互にデータ通信を実行する。

【0376】図49に戻りフローの説明を続ける。ステップS453，S463において上述のような相互認証、鍵共有処理に成功すると、デバイスは、ステップS464において、デバイスのメモリ部のデバイス鍵領域（図18参照）に格納されたCRL_DEV：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ex.シリアルナンバ：SN）を登録したリボケーションリスト（Revocation List (Certificate)）を参照して、通信相手であるデバイス認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0377】リボークされていない場合は、ステップS465において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手（デバイス認証装置を構成するデバイスアクセス機器としてのリーダライタ、PCなど）の公開鍵証明書中の識別名（DN：Distinguished Name）、シリアルナンバー、カテゴリーをデバイスマネージャコード（DMC）をキーとして対応付けた認証テーブルに保存する。

【0378】一方、デバイス認証装置も、ステップS454において、デバイスがリボークされていないかをCRL_DEV：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、PC等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ex.シリアルナンバ：SN）を登録したリボケーションリスト（Revocation List (Certificate)）を参照して判定する。デバイス認証装置は、リボケーションリスト（CRL_DEV）を登録局（RA（PAR））から取得可能である。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【0379】リボークされていない場合は、ステップS455において、相互認証および鍵共有処理において生成したセッション鍵Ksesと、通信相手（デバイス）の公開鍵証明書中の識別名（DN：Distinguished Name）、シリアルナンバー、カテゴリーをデバイスマネージャコード（DMC）をキーとして対応付けた認証テーブルに保存する。

【0380】図51にデバイス内に生成される認証テーブルの例を示し、図52に認証装置としてのデバイスアクセス機器としてのリーダライタ（PCも可）に生成される認証テーブルの例を示す。

【0381】図51は、デバイス認証、および後段で説明するパーティション認証としてのパーティション1，2の認証が終了した時点のデバイス内に生成される認証テーブルの例である。グループ（Group）として、デバイス認証の場合はデバイスマネージャコード（DMC）、パーティション認証の場合はパーティションマネージャコード（PMC）が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵Ksesと、通信相手（デバイスアクセス機器としてのリーダライタ）の公開鍵証明書中の識別名（DN：Distinguished Name）、シリアルナンバー、カテゴリーがテーブルに格納され、共通鍵認証の場合は、セッション鍵Ksesと、通信相手（デバイスアクセス機器としてのリーダライタ）の識別子（ID RW）ガ格納される。

【0382】認証テーブルはセッションのクリア時点で破棄される。デバイスはテーブルの情報を参照することによってデバイスおよび各パーティションの認証状態を確認することができ、使用すべきセッション鍵の確認が可能となる。

【0383】一方、図52は、デバイスアクセス機器としてのリーダライタ側の認証テーブルを示している。こ

の例もデバイス認証、およびパーティション認証としてのパーティション１，２の認証が終了した時点の認証テーブルの例である。基本的構成は、デバイス内の認証テーブルと同様であり、グループ（Group）として、デバイス認証の場合はデバイスマネージャコード（DMC）、パーティション認証の場合はパーティションマネージャコード（PMC）が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵Ｋｓｅｓと、通信相手（デバイス）の公開鍵証明書中の識別名（ＤＮ：Distinguished Name）、シリアルナンバー、カテゴリーがテーブルに格納され、共通鍵認証の場合は、セッション鍵Ｋｓｅｓと、通信相手（デバイス）の識別子（ＩＤ RW）が格納される。リーダライタ側の認証テーブルもセッションのクリア時点で破棄される。デバイスアクセス機器としてのリーダライタ側においても、認証テーブルの情報を参照することによってデバイスおよびパーティションの認証状態の有無を判定可能となり、使用すべきセッション鍵の確認が可能となる。

【０３８４】図４９に戻り、デバイス認証処理の説明を続ける。デバイス認証装置はステップＳ４５１において、デバイス認証方式が公開鍵方式でないと判定されると、デバイス認証装置はステップＳ４５６において、共通鍵デバイス認証コマンドをデバイスに出力する。デバイスは、コマンドを受信（Ｓ４６６）すると、デバイスのメモリ部の共通鍵系デバイス鍵定義ブロック（図１６参照）を参照して共通鍵認証に使用する双方向個別鍵認証用マスター鍵（MKauth__DEV）が格納されているか否かを検証（Ｓ４６７）する。双方向個別鍵認証用マスター鍵（MKauth__DEV）が格納されていない場合は、共通鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【０３８５】双方向個別鍵認証用マスター鍵（MKauth__DEV）が格納されていることが確認されると、ステップＳ４５７、Ｓ４６８において、マスター鍵を使用した相互認証および鍵共有処理が実行される。

【０３８６】マスター鍵を使用した共通鍵方式による相互認証および鍵共有処理について、図５３を用いて説明する。図５３では、AおよびBがマスター鍵を使用した共通鍵方式の認証を実行するエンティテイであり、Aは、自己の識別子ＩＤａ、双方向個別鍵認証用共通鍵Ｋａ、双方向個別鍵認証用マスター鍵ＭＫｂを有し、Bは、自己の識別子ＩＤｂ、双方向個別鍵認証用共通鍵Ｋｂ、双方向個別鍵認証用マスター鍵ＭＫａを有する。なお、図５３の例では、共通鍵暗号方式としてＤＥＳアルゴリズム（ｅｘ．ＤＥＳ，トリプルＤＥＳ）を用いているが、同様な共通鍵暗号方式であれば他の暗号方式の適用も可能である。

【０３８７】まず、Bが６４ビットの乱数Ｒｂを生成し、Ｒｂおよび自己のＩＤであるＩＤｂをAに送信する。これを受信したAは、新たに６４ビットの乱数Ｒａを生成し、双方向個別鍵認証用マスター鍵ＭＫｂによるＩＤｂのＤＥＳ暗号化処理により双方向個別鍵認証用共通鍵Ｋｂを取得する。さらに、鍵Ｋａ、Ｋｂを用いて、Ｒａ，Ｒｂ，ＩＤａ，ＩＤｂの順に、ＤＥＳのＣＢＣモードでデータを暗号化し、自己の識別子ＩＤａとともにBに返送する。

【０３８８】これを受信したBは、まず、双方向個別鍵認証用マスター鍵ＭＫａによるＩＤａのＤＥＳ暗号化処理により双方向個別鍵認証用共通鍵Ｋａを取得する。さらに、受信データを鍵Ｋａ，Ｋｂで復号する。復号して得られたＲａ、Ｒｂ、ＩＤａ，ＩＤｂの内、ＲｂおよびＩＤｂが、Bが送信したものと一致するか検証する。この検証に通った場合、BはAを正当なものとして認証する。

【０３８９】次にBは、セッション鍵として使用する６４ビットの乱数Ｋｓｅｓを生成し、鍵Ｋｂ、Ｋａを用いて、Ｒｂ，Ｒａ，ＩＤｂ，ＩＤａ，Ｋｓｅｓの順に、ＤＥＳのＣＢＣモードでデータを暗号化し、Aに返送する。

【０３９０】これを受信したAは、受信データを鍵Ｋａ，Ｋｂで復号する。復号して得られたＲａ、Ｒｂ、ＩＤａ，ＩＤｂがAが送信したものと一致するか検証する。この検証に通った場合、AはBを正当なものとして認証する。互いに相手を認証した後には、セッション鍵Ｋｓｅｓは、認証後の秘密通信のための共通鍵として利用される。

【０３９１】なお、受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を中止する。

【０３９２】本発明のシステムの格納データに対応付けたマスター鍵を使用した共通鍵認証について、データの流れを説明する図を図５４に示す。図５４に示すように、デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）が６４ビットの乱数Ｒｂを生成し、Ｒｂおよび自己のＩＤであるＩＤｒｗをデバイス（Ｄｅｖｉｃｅ）に送信する。これを受信したデバイス（Ｄｅｖｉｃｅ）は、新たに６４ビットの乱数Ｒａを生成し、双方向個別鍵認証用マスター鍵MKauth__DEV_AによるＩＤｒｗのＤＥＳ暗号化処理により双方向個別鍵認証用共通鍵Kauth__DEV_Aを取得する。さらに、鍵Kauth__DEV_A、Kauth__DEV_Bを用いて、Ｒａ，Ｒｂ，ＩＤｒｗの順に、暗号アルゴリズムとして、例えばＤＥＳ－ＣＢＣモードでデータを暗号化し、自己の識別子ＩＤｍとともにデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）に返送する。

【０３９３】これを受信したリーダライタ（Ｒ／Ｗ）は、まず、双方向個別鍵認証用マスター鍵MKauth__DEV_BによるＩＤｍのＤＥＳ暗号化処理により双方向個別鍵認証用共通鍵Kauth__DEV_Bを取得する。さらに、受信データを鍵Kauth__DEV_A、Kauth__DEV_Bで復号する。復号

して得られたＲｂおよびＩＤｒｗが、デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）が送信したものと一致するか検証する。この検証に通った場合、リーダライタ（Ｒ／Ｗ）はデバイス（Ｄｅｖｉｃｅ）を正当なものとして認証する。

【０３９４】次にリーダライタ（Ｒ／Ｗ）は、セッション鍵として使用する６４ビットの乱数Ｋｓｅｓを生成し、双方向個別鍵認証用共通鍵Kauth_DEV_A、 Kauth_DEV_Bを用いて、Ｒｂ，Ｒａ，Ｋｓｅｓの順に、ＤＥＳアルゴリズムとしての例えばトリプルＤＥＳモードでデータを暗号化し、デバイス（Ｄｅｖｉｃｅ）に返送する。

【０３９５】これを受信したデバイスは、受信データを双方向個別鍵認証用共通鍵Kauth_DEV_A、 Kauth_DEV_Bで復号する。復号して得られたＲａ、Ｒｂ，ＩＤｒｗがデバイス（Ｄｅｖｉｃｅ）が送信したものと一致するか検証する。この検証に通った場合、デバイス（Ｄｅｖｉｃｅ）はリーダライタ（Ｒ／Ｗ）を正当なものとして認証し、認証後、セッション鍵Ｋｓｅｓを秘密通信のための共通鍵として利用する。

【０３９６】なお、デバイスの固有値としてのＩＤｍは、先に図６のデバイスメモリフォーマットを使用して説明したようにデバイスマネージャ管理領域の格納データに基づく値を適用することができる。

【０３９７】上述したように、マスター鍵を使用した共通鍵方式による相互認証および鍵共有処理によれば、通信相手方のマスター鍵に基づく処理を実行して生成した通信相手の個別鍵と、自己の個別鍵の２つの鍵を共通鍵として設定し、設定した２つの鍵を用いて共通鍵方式による相互認証を実行する構成であるので、デバイスまたはアクセス装置に予め共通鍵が格納された従来の共通鍵認証構成に比較してよりセキュアな認証システムおよび方法が実現される。

【０３９８】図４９に戻りフローの説明を続ける。ステップＳ４５７，Ｓ４６８において上述のような相互認証、鍵共有処理に成功すると、デバイスは、ステップＳ４６９において、デバイスのメモリ部のデバイス鍵領域（図１８参照）に格納されたIRL_DEV：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List (ID)）を参照して、通信相手であるデバイス認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【０３９９】リボークされていない場合は、ステップＳ４７０において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（デバイス認証装置を構成するデバイスアクセス機器としてのリーダライタ、ＰＣなど）の識別情報（ＩＤｒｗ）をデバイスマネージャコード（ＤＭＣ）をキーとして対応付けた認証テーブル（図５１参照）に保存する。

【０４００】一方、デバイス認証装置も、ステップＳ４５８において、デバイスがリボークされていないかをIRL_DEV：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List (ID)）を参照して判定する。デバイス認証装置は、リボケーションリスト（IRL_DEV）を登録局（ＲＡ（ＰＡＲ））から取得可能である。リボークされている場合は、パーティションの生成処理を許可できないので、エラーとして処理を終了する。

【０４０１】リボークされていない場合は、ステップＳ４５９において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（デバイス）の識別情報（ＩＤｍ）をデバイスマネージャコード（ＤＭＣ）をキーとして対応付けた認証テーブル（図５２参照）に保存する。

【０４０２】以上の処理が、パーティションマネージャの管轄するデバイスアクセス機器としてのリーダライタとデバイス間において実行されるデバイス認証処理である。

【０４０３】次に、図４８のステップＳ４３５、Ｓ４４２において実行されるパーティション認証処理について、図５５、図５６を用いて説明する。パーティション登録チケットを適用したパーティションの設定登録、または削除処理においては、先に説明したように処理に応じてデバイス認証、パーティション認証のいずれか、または両者の認証を必要とする。これらの設定はパーティション登録チケット（ＰＲＴ）に記述される。パーティション登録チケット（ＰＲＴ）にパーティション認証を実行する記述がある場合は、パーティション認証が実行される。

【０４０４】図５５、図５６の処理フローの各ステップについて説明する。図５５において、左側がパーティションマネージャのパーティション認証装置、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャのパーティション認証装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する構成を有する。

【０４０５】パーティション認証装置は、ステップＳ４７１において、デバイスに対して認証対象となるパーティションＡの存在確認を実行するパーティションＡ存在チェックコマンドを出力する。コマンドを受領（Ｓ４８１）したデバイスは、デバイスのメモリ部内にパーティションＡが存在するか否かをチェック（Ｓ４８２）す

る。ここでパーティションの識別子Ａとしては例えばパーティションマネージャコード（ＰＭＣ）が使用され、デバイスは、例えばパーティション定義ブロック（ＰＤＢ：Partition Definition Block）の格納ＰＭＣに基づいてパーティションの有無を判定することができる。デバイスによるパーティションの有無が判定されると、チェック結果がパーティション認証装置に送信される。

【０４０６】チェック結果を受領（Ｓ４７２）したパーティション認証装置は、チェック結果を検証（Ｓ４７３）し、パーティションの存在しないとの結果を受領した場合は、認証不可であるので、エラー終了する。チェック結果がパーティションが存在することを示した場合は、パーティション認証装置は、ステップＳ４７４において、パーティション登録チケット（ＰＲＴ）に基づいてパーティション認証処理に公開鍵を用いた公開鍵認証方式を適用するか否かを判定する。パーティション認証は、前述のデバイス認証と同様、公開鍵方式または共通鍵方式のいずれかにおいて実行され、チケットにその実行方式についての指定が記述されている。

【０４０７】共通鍵方式の指定がある場合は、図５５のステップＳ４７５～Ｓ４７８、Ｓ４８４～Ｓ４８８の処理は実行されず、ステップＳ４９１に進む。公開鍵方式の指定である場合は、パーティション認証装置は、ステップＳ４７５において公開鍵パーティションＡ認証開始コマンドをデバイスに送信する。デバイスは、コマンドを受信（Ｓ４８４）すると、デバイスのメモリ部の公開鍵系パーティション鍵定義ブロック（図２１参照）を参照して、パーティションＡ対応公開鍵証明書（ＣＥＲＴ　ＰＡＲ）が格納されているか否かを検証（Ｓ４８５）する。パーティションＡ対応公開鍵証明書（ＣＥＲＴ　ＰＡＲ）が格納されていない場合は、公開鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【０４０８】パーティションＡ対応公開鍵証明書（ＣＥＲＴ　ＰＡＲ）が格納されていることが確認されると、ステップＳ４７６、Ｓ４８６において、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の発行した公開鍵証明書を用いた相互認証および鍵共有処理が実行される。

【０４０９】公開鍵方式による相互認証および鍵共有処理については、先のデバイス認証処理において説明した図５０に示すシーケンスと同様であるので説明を省略する。ただし、パーティション認証において利用する公開鍵証明書は、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の発行した公開鍵証明書であり、この公開鍵証明書の署名検証には、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵（ＰＵＢ　ＣＡ（ＰＡＲ））を用いる。公開鍵（ＰＵＢ　ＣＡ（ＰＡＲ））は、パーティション鍵領域（図２３参照）に格納されている。このような相互認証処理において、生成したセッション鍵を用いて、送信データを暗号化して、相互にデータ通信を実行する。

【０４１０】ステップＳ４７６，Ｓ４８６において図５０に示すシーケンスに従った相互認証、鍵共有処理に成功すると、デバイスは、ステップＳ４８７において、デバイスのメモリ部のパーティション鍵領域（図２３参照）に格納されたCRL_PAR：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate)を参照して、通信相手であるパーティション認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理あるいは削除処理を許可できないので、エラーとして処理を終了する。

【０４１１】リボークされていない場合は、ステップＳ４８８において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（パーティション認証装置を構成するデバイスアクセス機器としてのリーダライタ、ＰＣなど）の公開鍵証明書中の識別名（ＤＮ：Distinguished Name）、シリアルナンバー、カテゴリーをパーティションマネージャコード（ＰＭＣ）をキーとして対応付けた認証テーブルに保存する。

【０４１２】一方、パーティション認証装置も、ステップＳ４７７において、デバイスがリボークされていないかをCRL_PAR：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate））を参照して判定する。デバイス認証装置は、リボケーションリスト（CRL_PAR）を登録局（ＲＡ（ＰＡＲ））から取得可能である。リボークされている場合は、パーティションの生成処理あるいは削除処理を許可できないので、エラーとして処理を終了する。

【０４１３】リボークされていない場合は、ステップＳ４７８において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（デバイス）の公開鍵証明書中の識別名（ＤＮ：Distinguished Name）、シリアルナンバー、カテゴリーをパーティションマネージャコード（ＰＭＣ）をキーとして対応付けた認証テーブルに保存する。この結果、例えば図５１に示す認証テーブルがデバイス内に生成され、図５２に示す認証テーブルがパーティション認証装置としてのデバイスアクセス機器としてのリーダライタ（ＰＣも可）に生成される。図５１、図５２は、デバイス認証、およびパーティション認証としてのパーティション１，２の認証が終了した時点のデバイス内およびデバイスアクセス機器としてのリーダライタ内に生成される認証テーブルの例

である。

【０４１４】パーティション認証の場合はパーティションマネージャコード（ＰＭＣ）が記録され、実行された各認証方式によってそれぞれのデータが格納される。公開鍵認証方式の場合は、前述したようにセッション鍵Ｋｓｅｓと、通信相手の公開鍵証明書中の識別名（ＤＮ：Distinguished Name）、シリアルナンバー、カテゴリーがテーブルに格納され、共通鍵認証の場合は、セッション鍵Ｋｓｅｓと、通信相手の識別子が格納される。認証テーブルはセッションのクリア時点で破棄される。デバイスおよびデバイスアクセス機器としてのリーダライタはテーブルの情報を参照することによってデバイスおよび各パーティションの認証状態を確認することができ、使用すべきセッション鍵の確認が可能となる。

【０４１５】図５５，図５６のフローを用いて、パーティション認証処理の説明を続ける。パーティション認証装置はステップＳ４７４において、パーティション認証方式が公開鍵方式でないと判定されると、パーティション認証装置はステップＳ４９１において、共通鍵パーティションＡ認証コマンドをデバイスに出力する。デバイスは、コマンドを受信（Ｓ５０１）すると、デバイスのメモリ部の共通鍵系パーティション鍵定義ブロック（図２２参照）を参照して共通鍵認証に使用する双方向個別鍵認証用マスター鍵（MKauth__PAR_A）が格納されているか否かを検証（Ｓ５０２）する。双方向個別鍵認証用マスター鍵（MKauth__PAR_A）が格納されていない場合は、共通鍵方式の相互認証は実行できず、エラーと判定され処理は終了される。

【０４１６】双方向個別鍵認証用マスター鍵（MKauth__PAR_A）が格納されていることが確認されると、ステップＳ４９２、Ｓ５０３において、マスター鍵を使用した相互認証および鍵共有処理が実行される。共通鍵方式による相互認証および鍵共有処理については、先のデバイス認証において、図５３、図５４を用いて説明したシーケンスと同様であるので、説明を省略する。ただし、パーティション認証の場合に適用する鍵は、パーティション鍵定義ブロック（図２２参照）に定義され、パーティション鍵領域（図２３参照）に格納された双方向個別鍵認証用共通鍵（Kauth__PAR_B）、および双方向個別鍵認証用マスター鍵（MKauth__PAR_A）である。

【０４１７】ステップＳ４９２，Ｓ５０３において共通鍵方式の相互認証、鍵共有処理に成功すると、デバイスは、ステップＳ５０４において、デバイスのメモリ部のパーティション鍵領域（図２３参照）に格納されたIRL_PAR：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（ID））を参照して、通信相手であるパーティション認証装置がリボークされていないかを検証する。リボークされている場合は、パーティションの生成処理または削除処理を許可できないので、エラーとして処理を終了する。

【０４１８】リボークされていない場合は、ステップＳ５０５において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（デバイス認証装置を構成するデバイスアクセス機器としてのリーダライタ、ＰＣなど）の識別情報（ＩＤｒｗ）をパーティションマネージャコード（ＰＭＣ）をキーとして対応付けた認証テーブル（図５１参照）に保存する。

【０４１９】一方、パーティション認証装置も、ステップＳ４９３において、デバイスがリボークされていないかをＩRL_PAR：排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（ID））を参照して判定する。パーティション認証装置は、リボケーションリスト（IRL_PAR）を登録局（ＲＡ（ＰＡＲ））から取得可能である。リボークされている場合は、パーティションの生成処理または削除処理を許可できないので、エラーとして処理を終了する。

【０４２０】リボークされていない場合は、ステップＳ４９４において、相互認証および鍵共有処理において生成したセッション鍵Ｋｓｅｓと、通信相手（デバイス）の識別情報（ＩＤｍ）をパーティションマネージャコード（ＤＭＣ）をキーとして対応付けた認証テーブル（図５２参照）に保存する。

【０４２１】以上の処理が、パーティションマネージャの管轄するデバイスアクセス機器としてのリーダライタとデバイス間において実行されるパーティション認証処理である。このような相互認証により、デバイスまたはパーティションとデバイスアクセス機器としてのリーダライタ間の認証が成立し、セッション鍵の共有が達成され、通信データのセッション鍵による暗号化通信が可能となる。

【０４２２】なお、上述したデバイス認証処理、パーティション認証処理は、他のチケット、すなわちファイル登録チケット（ＦＲＴ：File Registration Ticket）、サービス許可チケット（ＳＰＴ：Service Permission Ticket）、データアップデートチケット（ＤＵＴ：Data Update Ticket）を使用したデバイスアクセスを実行する際にも適宜必要に応じて行われる処理である。これらについては、後段の各チケットを利用した処理の説明中で述べる。

【０４２３】（チケットの正当性と利用者チェック）次に、図４７のパーティションの作成、削除処理フロー中のステップＳ４１３のデバイスにおけるチケットの正当性と利用者チェック処理の詳細について図５７、図５８のフローを用いて説明する。

【０４２４】なお、以下に説明するチケットの正当性と利用者チェック処理は、他のチケット、すなわちファイ

ル登録チケット（ＦＲＴ：File Registration Ticket）、サービス許可チケット（ＳＰＴ：Service Permission Ticket）、データアップデートチケット（ＤＵＴ：Data Update Ticket）を使用したデバイスアクセス処理においても適宜必要に応じて行われる処理であり、図５７、図５８のフローは、各チケットに共通の処理フローとして構成してある。

【０４２５】チケットの正当性と利用者チェック処理は、デバイスとの通信を実行しているチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ等）から受信したチケットに基づいてデバイス（図５参照）が実行する処理である。デバイスは、チケットの正当性と利用者チェック処理においてチケットおよびチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ等）である利用者の正当性を確認した後、チケットに記述された制限範囲内の処理を許可する。

【０４２６】図５７、図５８のフローを用いてチケットの正当性と利用者チェック処理の詳細について説明する。チケットをチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ等）から受信したデバイスは、図５７のステップＳ５１１において、チケットタイプを検証しチケットがパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）であるか否かを判定する。チケットタイプは、各チケットに記録されている（図２６、図２７、図２８、図３１、図３２参照）。

【０４２７】チケットタイプがパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）である場合は、ステップＳ５１２～Ｓ５１４を実行し、パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）でない場合は、ステップＳ５１５に進む。

【０４２８】チケットタイプがパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）である場合は、ステップＳ５１２において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が公開鍵方式(Public)であるか否かを判定する。

【０４２９】正当性検証値の種別（Integrity Check Type）が公開鍵方式(Public)である場合、ステップＳ５１３に進み、各種処理を実行する。ステップＳ５１３で実行する処理は、まず、デバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の公開鍵ＰＵＢ ＣＡ（ＤＥＶ）を用いたチケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）の検証処理である。

【０４３０】前述したように、パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、パーティション登録チケット（ＰＲＴ）発行手段（PRT Issuer）の公開鍵証明書（CERT_PRTI）も一緒にデバイスに送信される。なお、ＰＲＴ発行手段の公開鍵証明書（CERT_PRTI）の属性（Attribute）は、パーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ User）の識別子(PRTIC)と一致する。

【０４３１】公開鍵証明書（図１１参照）にはデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の秘密鍵で実行された署名が付加されており、この署名をデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の公開鍵ＰＵＢ ＣＡ（ＤＥＶ）を用いて検証する。署名生成、検証は、例えば先に説明した図１２、図１３のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であるか否かが判定される。

【０４３２】さらに、ステップＳ５１３では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたユーザのカテゴリとしてのコードが、デバイス内のＤＫＤＢ（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）と一致するか否かを判定する。

【０４３３】公開鍵証明書には、図１１の公開鍵証明書の説明の欄で記述したように、各チケット（ＰＲＴ，ＦＲＴ、ＳＰＴなど）の発行手段であるチケット発行手段（Ticket Issuer）の所属コード、この場合、ＰＲＴＩＣ（PRT Issuer Code）が記録されている。このオプション領域のコードとデバイス内のＤＫＤＢ（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）の一致を確認することで、受信チケット（ＰＲＴ）が正当なチケット発行手段によって発行されたチケットであることを確認する。

【０４３４】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域（図１８参照）に格納されたCRL_DEV（排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate））を参照して、チケット発行手段（Ticket Issuer）がリボークされていないかを判定する。

【０４３５】さらに、受信チケットであるパーティション登録チケット（ＰＲＴ）（図２６参照）に記録された署名、すなわちIntegrity Check Value（チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature））））の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図１３のフローと同様のシーケンスに

従って実行される。

【０４３６】以上、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＤＫＤＢ（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＰＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてチケットの正当性検証成功とする。上記（１）～（４）のいずれかが確認されない場合は、チケットの正当性の確認が得られないと判定され、パーティション登録チケット（ＰＲＴ：Partition Registration Ticket）を利用した処理は中止される。

【０４３７】また、ステップＳ５１２において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が共通鍵方式(Common)であると判定された場合は、ステップＳ５１４に進みＭＡＣ（Message Authentication Code）検証を実行する。デバイスは、デバイスのデバイス鍵領域（図１８参照）に格納されたパーティション登録チケット（ＰＲＴ）のＭＡＣ検証用鍵：Ｋｐｒｔを使用してチケットのＭＡＣ検証処理を実行する。

【０４３８】図５９にＤＥＳ暗号処理構成を用いたＭＡＣ値生成例を示す。図５９の構成に示すように対象となるメッセージを８バイト単位に分割し、（以下、分割されたメッセージをＭ１、Ｍ２、・・・、ＭＮとする）、まず、初期値（Initial Value（以下、ＩＶとする））とＭ１を排他的論理和する（その結果をＩ１とする）。次に、Ｉ１をＤＥＳ暗号化部に入れ、ＭＡＣ検証用鍵：Ｋｐｒｔを用いて暗号化する（出力をＥ１とする）。続けて、Ｅ１およびＭ２を排他的論理和し、その出力Ｉ２をＤＥＳ暗号化部へ入れ、鍵Ｋｐｒｔを用いて暗号化する（出力Ｅ２）。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。最後に出てきたＥＮがメッセージ認証符号（ＭＡＣ（Message Authentication Code））となる。なお、メッセージとしては、検証対象となるデータを構成する部分データが使用可能である。

【０４３９】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶ（Integrity Check Value）と、データ受信側が受信データに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればデータに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶは、図２６のパーティション登録チケット（ＰＲＴ）のフォーマットに関する記述において説明したように、ＰＲＴのＩＣＶ（Integrity Check Value）フィールドに格納されている。デバイスが生成したＩＣＶと受信チケット（ＰＲＴ）に格納されたＩＣＶとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、チケットを利用した処理を中止する。

【０４４０】上述の処理によってチケットに記述されたIntegrity Check Typeが共通鍵方式である場合のチケット検証処理が完了する。

【０４４１】図５７のフローに戻り、チケットの正当性と利用者チェック処理について説明を続ける。ステップＳ５１１において、チケットタイプがパーティション登録チケット（ＰＲＴ：Partition Registration Ticket）でないと判定された場合は、ステップＳ５１５においてチケットタイプを検証しチケットがファイル登録チケット（ＦＲＴ：File Registration Ticket）であるか否かを判定する。

【０４４２】チケットタイプがファイル登録チケット（ＦＲＴ：File Registration Ticket）である場合は、ステップＳ５１６～Ｓ５１８を実行し、ファイル登録チケット（ＦＲＴ：File Registration Ticket）でない場合は、ステップＳ５１９に進む。

【０４４３】チケットタイプがファイル登録チケット（ＦＲＴ：File Registration Ticket）である場合は、ステップＳ５１６において、チケットに記述されたIntegrityCheck Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が公開鍵方式(Public)であるか否かを判定する。

【０４４４】正当性検証値の種別（Integrity Check Type）が公開鍵方式(Public)である場合、ステップＳ５１７に進み、各種処理を実行する。ステップＳ５１７で実行する処理は、まず、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いたチケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）の検証処理である。

【０４４５】ファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）の公開鍵証明書（CERT_FRTI）も一緒にデバイスに送信される。なお、ＦＲＴ発行手段の公開鍵証明書（CERT_FRTI）の属性（Attribute）は、、ファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）の識別子(FRTIC)と一致する。

【０４４６】公開鍵証明書（図１１参照）にはパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の秘密鍵で実行された署名が付加されており、この署名をパー

ティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いて検証する。署名生成、検証は、例えば先に説明した図１２、図１３のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であるか否かが判定される。

【０４４７】さらに、ステップＳ５１７では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたユーザの所属コードと、デバイス内のＰＫＤＢ（Partition Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＦＲＴＩＣ：FRT Issuer Code）と一致するか否かを判定する。

【０４４８】公開鍵証明書には、図１１の公開鍵証明書の説明の欄で記述したように、各チケット（ＰＲＴ，ＦＲＴ、ＳＰＴなど）の発行手段であるチケット発行手段（Ticket Issuer）の所属コード、この場合、ＦＲＴＩＣ（FRT Issuer Code）が記録されている。このオプション領域のコードとデバイス内のＰＫＤＢ（PartitionKey Definition Block）(PUB)に記録されたチケット発行手段コード（ＦＲＴＩＣ：FRT Issuer Code）の一致を確認しすることで、受信チケット（ＦＲＴ）が正当なチケット発行手段によって発行されたチケットであることを確認する。

【０４４９】さらに、デバイスは、デバイスのメモリ部内のパーティション鍵領域（図２３参照）に格納されたCRL_PAR（排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate））を参照して、チケット発行手段（Ticket Issuer）がリボークされていないかを判定する。

【０４５０】さらに、受信チケットであるファイル登録チケット（ＦＲＴ）（図２７参照）に記録された署名、すなわちIntegrity Check Value（チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature））の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図１３のフローと同様のシーケンスに従って実行される。

【０４５１】以上、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＰＫＤＢ（Partition Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＦＲＴＩＣ：FRT Issuer Code）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＦＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてファイル登録チケット（ＦＲＴ）の正当性検証成功とする。上記（１）～（４）のいずれかが確認されない場合は、ファイル登録チケット（ＦＲＴ）の正当性の確認が得られないと判定され、ファイル登録チケット（ＦＲＴ）を利用した処理は中止される。

【０４５２】また、ステップＳ５１６において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が共通鍵方式(Common)であると判定された場合は、ステップＳ５１８に進みＭＡＣ（Message Authentication Code）検証を実行する。デバイスは、デバイスのパーティション鍵領域（図２３参照）に格納されたファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵：Ｋｆｒｔを使用してチケットのＭＡＣ検証処理を実行する。ＭＡＣ検証処理は、先に説明した図５９のＤＥＳ暗号処理構成を用いたＭＡＣ値生成処理に従って実行される。

【０４５３】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶ（Integrity Check Value）と、データ受信側が受信データに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればデータに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶは、図２７のファイル登録チケット（ＦＲＴ）のフォーマットに関する記述において説明したように、ＦＲＴのＩＣＶ（Integrity Check Value）フィールドに格納されている。デバイスが生成したＩＣＶと受信チケット（ＦＲＴ）に格納されたＩＣＶとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、チケットを利用した処理を中止する。

【０４５４】上述の処理によってチケットに記述されたIntegrity Check Typeが共通鍵方式である場合のファイル登録チケット（ＦＲＴ）検証処理が完了する。

【０４５５】ステップＳ５１５において、チケットタイプがファイル登録チケット（ＦＲＴ：File Registration Ticket）でないと判定された場合は、ステップＳ５１９においてチケットタイプを検証しチケットがサービス許可チケット（ＳＰＴ：Service Permission Ticket）であるか否かを判定する。

【０４５６】チケットタイプがサービス許可チケット（ＳＰＴ：Service Permission Ticket）である場合は、ステップＳ５２０～Ｓ５２２を実行し、サービス許可チケット（ＳＰＴ：Service Permission Ticket）でない場合は、ステップＳ５２３に進む。

【０４５７】チケットタイプがサービス許可チケット（ＳＰＴ：Service Permission Ticket）である場合は、ステップＳ５２０において、チケットに記述されたIntegrityCheck Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が公開鍵方式(Public)であるか否かを判定する。

【０４５８】正当性検証値の種別（Integrity Check Type）が公開鍵方式(Public)である場合、ステップＳ５２１に進み、各種処理を実行する。ステップＳ５２１で実行する処理は、まず、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いたチケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）の検証処理である。

【０４５９】サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット（ＳＲＴ）発行手段（SPT Issuer）の公開鍵証明書（CERT_SPTI）も一緒にデバイスに送信される。なお、ＳＰＴ発行手段の公開鍵証明書（CERT_SPTI）の属性（Attribute）は、サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の識別子(SPTIC)と一致する。

【０４６０】公開鍵証明書（図１１参照）にはパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の秘密鍵で実行された署名が付加されており、この署名をパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いて検証する。署名生成、検証は、例えば先に説明した図１２、図１３のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であるか否かが判定される。

【０４６１】さらに、ステップＳ５２１では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたユーザの所属コードと、デバイス内のファイル定義ブロック（ＦＤＢ：File Definition Block）に記録されたチケット発行手段コード（ＳＰＴＩＣ：SPT Issuer Code）と一致するか否かを判定する。

【０４６２】公開鍵証明書には、図１１の公開鍵証明書の説明の欄で記述したように、各チケット（ＰＲＴ，ＦＲＴ、ＳＰＴなど）の発行手段であるチケット発行手段（Ticket Issuer）の所属コード、この場合、ＳＰＴＩＣ（SPT Issuer Code）が記録されている。このオプション領域のコードとデバイス内のＦＤＢ（File Definition Block）に記録されたチケット発行手段コード（ＳＰＴＩＣ：SPT Issuer Code）の一致を確認しすることで、受信チケット（ＳＰＴ）が正当なチケット発行手段によって発行されたチケットであることを確認する。

【０４６３】さらに、デバイスは、デバイスのメモリ部内のパーティション鍵領域（図２３参照）に格納されたCRL_PAR（排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)））を参照して、チケット発行手段（Ticket Issuer）がリボークされていないかを判定する。

【０４６４】さらに、受信チケットであるサービス許可チケット（ＳＰＴ）（図２８、図３１参照）に記録された署名、すなわちIntegrity Check Value（チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature））の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図１３のフローと同様のシーケンスに従って実行される。

【０４６５】以上、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＦＤＢ（File Definition Block）に記録されたチケット発行手段コード（ＳＰＴＩＣ：SPT Issuer Code）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＳＰＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてサービス許可チケット（ＳＰＴ）の正当性検証成功とする。上記（１）～（４）のいずれかが確認されない場合は、サービス許可チケット（ＳＰＴ）の正当性の確認が得られないと判定され、サービス許可チケット（ＳＰＴ）を利用した処理は中止される。

【０４６６】また、ステップＳ５２０において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が共通鍵方式(Common)であると判定された場合は、ステップＳ５２２に進みＭＡＣ（Message Authentication Code）検証を実行する。デバイスは、デバイスのファイル定義ブロック（図２４参照）に格納されたサービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵：Ｋｓｐｔを使用してチケットのＭＡＣ検証処理を実行する。ＭＡＣ検証処理は、先に説明した図５９のＤＥＳ暗号処理構成を用いたＭＡＣ値生成処理に従って実行される。

【０４６７】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶ（Integrity Check Value）と、データ受信側が受信データに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られれば

データに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶは、図２８、図３１のサービス許可チケット（ＳＰＴ）のフォーマットに関する記述において説明したように、ＳＰＴのＩＣＶ（Integrity Check Value）フィールドに格納されている。デバイスが生成したＩＣＶと受信チケット（ＳＰＴ）に格納されたＩＣＶとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、サービス許可チケット（ＳＰＴ）を利用した処理を中止する。

【０４６８】上述の処理によってサービス許可チケット（ＳＰＴ）に記述されたIntegrityCheck Typeが共通鍵方式である場合のサービス許可チケット（ＳＰＴ）検証処理が完了する。

【０４６９】ステップＳ５１９において、チケットタイプがサービス許可チケット（ＳＰＴ：Service Permission Ticket）でないと判定された場合は、ステップＳ５２３においてチケットタイプを検証しチケットがデータアップデートチケット-ＤＥＶ（ＤＵＴ：Data Update Ticket(DEV)）（図３２参照）であるか否かを判定する。データアップデートチケット（ＤＵＴ）は前述したようにデバイスのメモリ部に格納された各種データの更新処理を実行する際のアクセス許可チケットであり、デバイスマネージャの管理データを更新する処理に適用するデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））とパーティションマネージャの管理データを更新する処理に適用するデータアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））とがある。

【０４７０】チケットタイプがデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））である場合は、ステップＳ５２４～Ｓ５２８を実行し、データアップデートチケット（ＤＥＶ）（ＤＵＴ：Data Update Ticket(DEV)）でない場合は、ステップＳ５２９に進む。

【０４７１】チケットタイプがデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））である場合は、ステップＳ５２４において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が公開鍵方式(Public)であるか否かを判定する。

【０４７２】正当性検証値の種別（Integrity Check Type）が公開鍵方式(Public)である場合、ステップＳ５２５に進み、各種処理を実行する。ステップＳ５２５で実行する処理は、まず、デバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の公開鍵ＰＵＢ ＣＡ（ＤＥＶ）を用いたチケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）の検証処理である。

【０４７３】データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））発行手段（DUT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、データアップデートチケット（ＤＵＴ）発行手段（DUTIssuer）の公開鍵証明書（CERT_DUTI）も一緒にデバイスに送信される。なお、ＤＵＴ発行手段の公開鍵証明書（CERT_DUTI）の属性（Attribute）は、デバイス内のＤＫＤＢ（ＰＵＢ）（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（DUTIC_DEV）の識別子(DUTIC)と一致する。

【０４７４】公開鍵証明書（図１１参照）にはデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の秘密鍵で実行された署名が付加されており、この署名をデバイスマネージャ対応認証局（ＣＡ（ＤＥＶ））の公開鍵ＰＵＢ ＣＡ（ＤＥＶ）を用いて検証する。署名生成、検証は、例えば先に説明した図１２、図１３のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書（ＣＥＲＴ）が改竄されたものではない正当な公開鍵証明書（ＣＥＲＴ）であるか否かが判定される。

【０４７５】さらに、ステップＳ５２５では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたユーザの所属コードと、デバイス内のＤＫＤＢ（ＰＵＢ）（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（DUTIC_DEV：DUT Issuer Category for Device）と一致するか否かを判定する。

【０４７６】公開鍵証明書には、図１１の公開鍵証明書の説明の欄で記述したように、各チケット（ＰＲＴ，ＦＲＴ、ＳＰＴ、ＤＵＴ）の発行手段であるチケット発行手段（Ticket Issuer）の所属コード、この場合、ＤＵＴＩＣ（DUT Issuer Code）が記録されている。このオプション領域のコードとデバイス内のＤＫＤＢ（ＰＵＢ）（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（DUTIC_DEV：DUT Issuer Category for Device）（図１６参照）の一致を確認することで、受信チケット（ＤＵＴ）が正当なチケット発行手段によって発行されたチケットであることを確認する。

【０４７７】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域（図１８参照）に格納されたCRL_DEV（排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)））を参照して、チケット発行手段（Ticket Issuer）がリボークされていないかを判定する。

【０４７８】さらに、受信チケットであるデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））（図３２参照）に記録された署名、すなわちIntegrity Check Value（チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature））の検証を実行し、チケットが

改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図１３のフローと同様のシーケンスに従って実行される。

【０４７９】以上、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＤＫＤＢ（ＰＵＢ）（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（DUTIC_DEV：DUT Issuer Category for Device）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＤＵＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））の正当性検証成功とする。上記（１）～（４）のいずれかが確認されない場合は、データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））の正当性の確認が得られないと判定され、データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））を利用した処理は中止される。

【０４８０】また、ステップＳ５２４において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が共通鍵方式(Common)であると判定された場合は、ステップＳ５２６において、データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））に記述されたOld Data Codeの示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_DEV2（データ更新用暗号鍵）であるか否かを判定する。

【０４８１】データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_DEV2（データ更新用暗号鍵）である場合は、ステップＳ５２８において、デバイス鍵領域（図１８参照）に格納されたKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）を用いてＭＡＣ検証処理を実行し、データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_DEV2（データ更新用暗号鍵）でない場合は、ステップＳ５２７において、デバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）を用いてＭＡＣ検証処理を実行する。

【０４８２】上述のようにＭＡＣ検証鍵の使い分けを実行するのは、更新対象となっているデータが、Kdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_DEV2（データ更新用暗号鍵）である場合は、これらの鍵データが何らかの理由、例えば鍵情報の漏洩等により、使用を停止される予定の情報であるため、これらの更新対象データを用いたＭＡＣ検証を避けるためである。ＭＡＣ検証処理は、先に説明した図５９のＤＥＳ暗号処理構成を用いたＭＡＣ値生成処理に従って実行される。

【０４８３】なお、デバイスは、デバイスのデバイス鍵領域（図１８参照）に新規にKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）を格納する場合、以前に格納済みのKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）とのスワップ、すなわち入れ替え処理を行なう。さらに、新規にKdut_DEV2（データ更新用暗号鍵）を格納する場合も、以前に格納済みのKdut_DEV4（データ更新用暗号鍵）とのスワップ、すなわち入れ替え処理を行なう。

【０４８４】この、Kdut_DEV1と、Kdut_DEV3とのスワップ、および、Kdut_DEV2と、Kdut_DEV4とのスワップ処理によって、常にKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV4（データ更新用暗号鍵）のペアがKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）のペアよりも新しいバージョンのものに維持される。つまり、Kdut_DEV1と、Kdut_DEV2の鍵は常に使用される鍵で、Kdut_DEV3と、Kdut_DEV4は、非常時にKdut_DEV1と、Kdut_DEV2を更新するとともに、現在使用されているKdut_DEV1と、Kdut_DEV2の鍵に置き換えられるバックアップ用の鍵としての役割がある。なお、これらの処理については、後段のデータアップデートチケット（ＤＵＴ）を用いたデータ更新処理の説明において、さらに説明する。

【０４８５】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶ（Integrity Check Value）と、データ受信側が受信データに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればデータに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶは、図３２のデータアップデートチケット（ＤＵＴ）のフォーマットに関する記述において説明したように、データアップデートチケット（ＤＵＴ）のＩＣＶ（Integrity Check Value）フィールドに格納されている。

【０４８６】デバイスが生成したＩＣＶと受信チケットであるデータアップデートチケット-ＤＥＶ（ＤＵＴ

（ＤＥＶ））に格納されたＩＣＶとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、データアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））を利用した処理を中止する。

【０４８７】上述の処理によってデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））に記述されたIntegrity Check Typeが共通鍵方式である場合のデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））検証処理が完了する。

【０４８８】ステップＳ５２３において、チケットタイプがデータアップデートチケット-ＤＥＶ（ＤＵＴ（ＤＥＶ））でないと判定された場合は、チケットは、データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））(図３２参照))であると判定される。データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））は、パーティションマネージャの管理データを更新する処理に適用するチケットである。

【０４８９】この場合、ステップＳ５２９において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が公開鍵方式(Public)であるか否かを判定する。

【０４９０】正当性検証値の種別（Integrity Check Type）が公開鍵方式(Public)である場合、ステップＳ５３０に進み、各種処理を実行する。ステップＳ５３０で実行する処理は、まず、パーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いたチケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）の検証処理である。

【０４９１】データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））発行手段（DUT Issuer）の発行したチケット（Ticket）を、チケットユーザに対して送信する際には、公開鍵方式の場合、データアップデートチケット（ＤＵＴ）発行手段（DUTIssuer）の公開鍵証明書（CERT_DUTI）も一緒にデバイスに送信される。なお、ＤＵＴ発行手段の公開鍵証明書（CERT_DUTI）の属性（Attribute）は、デバイス内のＰＫＤＢ（ＰＵＢ）（Pqrtition Key Definition block）に記録されたチケット発行手段コード(DUTIC_PAR)と一致する。

【０４９２】公開鍵証明書（図１１参照）にはパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の秘密鍵で実行された署名が付加されており、この署名をパーティションマネージャ対応認証局（ＣＡ（ＰＡＲ））の公開鍵ＰＵＢ ＣＡ（ＰＡＲ）を用いて検証する。署名生成、検証は、例えば先に説明した図１２、図１３のフローに従った処理として実行される。この署名検証により、チケット発行者の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であるか否かが判定される。

【０４９３】さらに、ステップＳ５３０では、署名検証により正当性が確認されたチケット発行手段の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたユーザの所属コードと、デバイス内のＰＫＤＢ（ＰＵＢ）（Pqrtition Key Definition block）に記録されたチケット発行手段コード(DUTIC_PAR:DUT Issuer Category forPartition)と一致するか否かを判定する。

【０４９４】公開鍵証明書には、図１１の公開鍵証明書の説明の欄で記述したように、各チケット（ＰＲＴ，ＦＲＴ、ＳＰＴ、ＤＵＴ）の発行手段であるチケット発行手段（Ticket Issuer）の所属コード、この場合、ＤＵＴＩＣ（DUT Issuer Code）が記録されている。このオプション領域のコードとデバイス内のＰＫＤＢ（ＰＵＢ）（Pqrtition Key Definition block）に記録されたチケット発行手段コード(DUTIC:DUT Issuer Category)(図２１参照)の一致を確認しすることで、受信チケット（ＤＵＴ）が正当なチケット発行手段によって発行されたチケットであることを確認する。

【０４９５】さらに、デバイスは、デバイスのメモリ部内のデバイス鍵領域（図１８参照）に格納されたCRL_DEV（排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)））を参照して、チケット発行手段（Ticket Issuer）がリボークされていないかを判定する。

【０４９６】さらに、受信チケットであるデータアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））（図３２参照）に記録された署名、すなわちIntegrity Check Value（チケット（Ticket）の正当性検証値（公開鍵方式：署名（Signature））の検証を実行し、チケットが改竄されていないかを確認する。署名検証は、先の公開鍵証明書の署名検証と同様、例えば図１３のフローと同様のシーケンスに従って実行される。

【０４９７】以上、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、
（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＰＫＤＢ（ＰＵＢ）（Pqrtition Key Definition block）に記録されたチケット発行手段コード(DUTIC_PAR:DUT Issuer Category for Partition)の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＤＵＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認。以上のすべての確認がなされたことを条件としてデータアップデートチケット-ＰＡＲ（ＤＵＴ）の正当性検証成功とする。上記（１）～（４）のいずれかが確認されない場合は、データアップデートチ

ケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））の正当性の確認が得られないと判定され、データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））を利用した処理は中止される。

【０４９８】また、ステップＳ５２９において、チケットに記述されたIntegrity Check Type（チケット（Ticket）の正当性検証値の種別（公開鍵方式(Public)/共通鍵方式(Common)））の設定が共通鍵方式(Common)であると判定された場合は、ステップＳ５３１において、データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_PAR2（データ更新用暗号鍵）であるか否かを判定する。

【０４９９】データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_PAR2（データ更新用暗号鍵）である場合は、ステップＳ５３３において、パーティション鍵領域（図２３参照）に格納されたKdut_PAR3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）を用いてＭＡＣ検証処理を実行し、データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_PAR2（データ更新用暗号鍵）でない場合は、ステップＳ５３２において、パーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）を用いてＭＡＣ検証処理を実行する。

【０５００】上述のようにＭＡＣ検証鍵の使い分けを実行するのは、更新対象となっているデータが、Kdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_PAR2（データ更新用暗号鍵）である場合は、これらの鍵データが何らかの理由、例えば鍵情報の漏洩等により、使用を停止される予定の情報であるため、これらの更新対象データを用いたＭＡＣ検証を避けるためである。ＭＡＣ検証処理は、先に説明した図５９のＤＥＳ暗号処理構成を用いたＭＡＣ値生成処理に従って実行される。

【０５０１】改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶ（Integrity Check Value）と、データ受信側が受信データに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればデータに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。改竄のないことが保証された例えばデータ送信側がデータ生成時に生成したＩＣＶは、図３２のデータアップデートチケット（ＤＵＴ）のフォーマットに関する記述において説明したように、データアップデートチケット（ＤＵＴ）のＩＣＶ（Integrity Check Value）フィールドに格納されている。

【０５０２】デバイスが生成したＩＣＶと受信チケットであるデータアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））に格納されたＩＣＶとを比較して一致していればチケットの正当性ありと判定し、不一致の場合はチケット改竄ありと判定し、データアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））を利用した処理を中止する。

【０５０３】上述の処理によってデータアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））に記述されたIntegrity Check Typeが共通鍵方式である場合のデータアップデートチケット-ＰＡＲ（ＤＵＴ（ＰＡＲ））検証処理が完了する。

【０５０４】以上の処理においてチケットの正当性が確認された後、図５８のステップＳ５４１に進み、以下、利用者チェック、すなわちチケットユーザとしてデバイスとの通信を実行中のデバイスアクセス機器としてのリーダライタ（またはＰＣ等）のチェックを実行する。

【０５０５】ステップＳ５４１において、デバイスは、受信チケット（ＰＲＴ，ＦＲＴ，ＳＰＴ，またはＤＵＴ）のAuthentication Flag（チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ）をチェックする。フラグが認証不要を示している場合は、処理を実行することなく終了する。

【０５０６】ステップＳ５４１におけるフラグチェック処理において、フラグが認証必要を示している場合は、ステップＳ５４２に進み、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）の所属（グループ）をキーとして認証テーブル（図５１参照）を参照する。

【０５０７】次に、ステップＳ５４３において、受信チケットのAuthentication Type（デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録したデータ）をチェックし、いずれでも可（Any）である場合、ステップＳ５４４に進み、ステップＳ５４２でチェックしたグループの相互認証データが認証テーブル（図５１参照）に格納されているか否かを判定する。テーブルに対応グループの相互認証情報が格納され、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）とデバイス間の相互認証済みであることが判定されれば、チケット利用者（ｅｘ．デバイスアクセス機器と

してのリーダライタ）の正当性が確認されたものとして処理を利用者チェック成功と判定して終了する。認証テーブル（図５１参照）に対応グループの相互認証情報が格納されていない場合は、利用者チェックが未了であると判定され、エラー終了とする。

【０５０８】ステップＳ５４３において、受信チケットのAuthentication Type（デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録したデータ）がいずれでも可（Any）でない場合、ステップ５４５において、Authentication Typeが公開鍵認証であるか否かを判定する。

【０５０９】Authentication Typeが公開鍵認証である場合、ステップＳ５４６に進み、ステップＳ５４２でチェックしたグループの公開鍵相互認証データが認証テーブル（図５１参照）に格納されているか否かを判定する。テーブルに対応グループの公開鍵相互認証情報が格納され、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）とデバイス間の相互認証が公開鍵認証処理として成立済みであることが判定された場合は、ステップＳ５４７に進み、処理対象チケット（ＰＲＴ，ＦＲＴ，ＳＰＴまたはＤＵＴ）にチケットユーザの識別子が存在するか否かを判定して存在する場合は、ステップＳ５４８において認証相手（チケットユーザであるデバイスアクセス機器としてのリーダライタなど）の公開鍵証明書中の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）とチケットに格納されたチケットユーザの識別データとして記録された識別子またはカテゴリまたはシリアル（ＳＮ）が一致するか否かを判定する。一致する場合は、利用者確認成功として処理を終了する。

【０５１０】ステップＳ５４６において、ステップＳ５４２でチェックしたグループの公開鍵相互認証データが認証テーブル（図５１参照）に格納されておらず、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）とデバイス間の相互認証が公開鍵認証処理として成立済みでないと判定された場合は、利用者チェック未了と判定されエラー終了する。

【０５１１】また、ステップＳ５４８において認証相手（チケットユーザであるデバイスアクセス機器としてのリーダライタなど）の公開鍵証明書中の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）とチケットに格納されたチケットユーザの識別子が一致しないと判定された場合も利用者チェック未了と判定されエラー終了する。

【０５１２】なお、チケットにチケットユーザの識別子が存在しない場合は、ステップＳ５４８の処理は実行せず、利用者確認成功として処理を終了する。

【０５１３】ステップＳ５４５において、受信チケットのAuthentication Type（デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））を記録したデータ）が公開鍵認証でないと判定された場合、ステップＳ５４９に進み、ステップＳ５４２でチェックしたグループの共通鍵相互認証データが認証テーブル（図５１参照）に格納されているか否かを判定する。テーブルに対応グループの共通鍵相互認証情報が格納され、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）とデバイス間の相互認証が共通鍵認証処理として成立済みであることが判定された場合は、ステップＳ５５０に進み、処理対象チケット（ＰＲＴ，ＦＲＴ，ＳＰＴまたはＤＵＴ）にチケットユーザの識別子が存在するか否かを判定して存在する場合は、ステップＳ５５１において認証相手（チケットユーザであるデバイスアクセス機器としてのリーダライタなど）の識別データ（ＩＤｒｗ）とチケットに格納されたチケットユーザの識別子が一致するか否かを判定する。一致する場合は、利用者確認成功として処理を終了する。

【０５１４】ステップＳ５４９において、ステップＳ５４２でチェックしたグループの共通鍵相互認証データが認証テーブル（図５１参照）に格納されておらず、チケットユーザ（デバイスに対するチケットを適用した処理を実行しようとするデバイスアクセス機器としてのリーダライタ、ＰＣ等）とデバイス間の相互認証が共通鍵認証処理として成立済みでないと判定された場合は、利用者チェック未了と判定されエラー終了する。

【０５１５】また、ステップＳ５５１において認証相手（チケットユーザであるデバイスアクセス機器としてのリーダライタなど）の識別データ（ＩＤｒｗ）とチケットに格納されたチケットユーザの識別子が一致しないと判定された場合も利用者チェック未了と判定されエラー終了する。

【０５１６】なお、チケットにチケットユーザの識別子が存在しないか、すべてのチケットユーザが利用可能の場合は、ステップＳ５５０の処理は実行せず、利用者確認成功として処理を終了する。

【０５１７】以上が、図４７のフロー中のステップＳ４１３においてデバイスが実行するチケットの正当性および利用者チェック処理である。

【０５１８】（パーティション作成削除処理）次に、図４７のフローに示すステップＳ４１５において実行されるパーティション登録チケット（ＰＲＴ）に基づくパーティションの生成、削除処理の詳細について、図６０、図６１の処理フローを用いて説明する。パーティションの作成、削除処理は、チケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ等）からパーティション登録チケット（ＰＲＴ）を受信したデバイス

が、パーティション登録チケット（ＰＲＴ）に基づいて実行する処理である。

【０５１９】図６０のステップＳ６０１において、デバイスは、受信したパーティション登録チケット（ＰＲＴ:Partition Registration ticket）に記録された処理タイプ、すなわちOperation Type（パーティション（Partition）作成か削除かの指定(作成（Generate）／削除（Delete）））を検証する。処理タイプ（Operation Type）が、パーティション（Partition）作成である場合、ステップＳ６０２以下を実行し、パーティション（Partition）削除である場合、ステップＳ６２１以下を実行する。

【０５２０】まず、パーティション作成処理について説明する。デバイスはステップＳ６０２において、パーティション登録チケット（ＰＲＴ）に記述されたパーティションマネージャコード（ＰＭＣ）と同一コードのパーティションがデバイスのメモリ部に存在するか否かを検証する。この判定は、デバイスのメモリ部のパーティション定義ブロック（図１９参照）に受信チケット（ＰＲＴ）の記述コードと同一のコードが記述されているか否かを検証することによって判定可能である。

【０５２１】すでにデバイスに同一コード（ＰＭＣ）のパーティションが存在する場合は、同一コードを持つ重複パーティションの存在は許されず、パーティションの生成は実行せず、エラー終了とする。同一コードのパーティションがデバイスに存在しない場合は、ステップＳ６０３において、デバイス管理情報ブロック（図１５参照）のデバイス（Device）内の空きブロック数（Free Block Number in Device）と、パーティション登録チケット（ＰＲＴ）に記述されたパーティションサイズ（Partion Size）とを比較し、チケット（ＰＲＴ）に記述されたパーティションサイズ（Partion Size）以上の空きブロック領域がデバイスのメモリ部に存在するか否かを判定する。存在しない場合は、ＰＲＴに記述されたサイズのパーティションの生成はできないので、エラー終了とする。

【０５２２】チケット（ＰＲＴ）に記述されたパーティションサイズ（Partion Size）以上の空きブロック領域がデバイスのメモリ部に存在すると判定された場合は、ステップＳ６０４に進み、デバイス管理情報ブロックの空き領域ポインタ（Pointerof Free Area）を参照してデバイスの空き領域（Free Area in Device）の最上位ブロックにパーティション定義ブロック（ＰＤＢ）エリア（図１９参照）を確保する。

【０５２３】次に、デバイスは、確保したパーティション定義ブロック（ＰＤＢ）エリアに、パーティション登録チケット（ＰＲＴ）に記述されたパーティションマネージャコード（ＰＭＣ）のコピー（Ｓ６０５）、ＰＲＴに記述されたＰＭＣバージョンのコピー、（Ｓ６０６）を実行する。

【０５２４】さらに、パーティション定義ブロック（ＰＤＢ）エリアのパーティションスタート位置（Partition Start Position）に、デバイス管理情報ブロック（図１５参照）の空き領域ポインタ（Pointer of Free Area）のコピー処理を実行（Ｓ６０７）し、さらに、パーティション定義ブロック（ＰＤＢ）エリアのパーティションサイズ（Partion Size）にパーティション登録チケット（ＰＲＴ）に記述されたパーティションサイズ（Partion Size）のコピー処理を実行（Ｓ６０８）する。

【０５２５】次に、デバイス管理情報ブロック（図１５参照）の空き領域ポインタ（Pointer of Free Area）にパーティション定義ブロック（ＰＤＢ）エリアのパーティションサイズ（Partion Size）にコピーした値を加算（Ｓ６０９）し、デバイス管理情報ブロック（図１５参照）のデバイス（Device）内の空きブロック数（FreeBlock Number in Device）からパーティションサイズ（Partion Size）＋１を減算する（Ｓ６１０）。なお、＋１は、パーティション定義ブロック（ＰＤＢ）用のブロックを意味する。

【０５２６】次にデバイス管理情報ブロック（図１５参照）のパーティション数（Partition Number）に１を加算、すなわち生成したパーティション数（１）を加算する（Ｓ６１１）。

【０５２７】次に、図６１のステップＳ６３１において、生成したパーティション領域の最上位ブロックをパーティション管理情報ブロック（ＰＭＩＢ：partition Management Information Block）（図２０参照）として設定し、設定したパーティション管理情報ブロック（ＰＭＩＢ）のパーティションマネージャコード（ＰＭＣ）フィールドにパーティション登録チケット（ＰＲＴ）のＰＭＣのコピー処理を実行（Ｓ６３２）し、パーティション管理情報ブロック（ＰＭＩＢ）のＰＭＣバージョンフィールドにパーティション登録チケット（ＰＲＴ）のＰＭＣバージョンのコピー処理を実行（Ｓ６３３）し、パーティション管理情報ブロック（ＰＭＩＢ）のパーティション総ブロック数（Total Block number in Partition）フィールドにパーティション登録チケット（ＰＲＴ）のパーティションサイズ（Partion Size）のコピー処理を実行（Ｓ６３４）する。

【０５２８】さらに、パーティション管理情報ブロック（ＰＭＩＢ）のパーティション空きブロック数（Free Block number in Partition）フィールドにパーティション登録チケット（ＰＲＴ）のパーティションサイズ（Partion Size）－３を記録（Ｓ６３５）する。なお、－３の意味は、既に使用が予定されているパーティション管理情報ブロック（ＰＭＩＢ）、共通鍵系パーティション鍵定義ブロック（ＰＫＤＢ(common)）、公開鍵系パーティション鍵定義ブロック（ＰＫＤＢ(PUB)）の３ブロックを差し引くことを意味している。

【０５２９】さらに、パーティション管理情報ブロック

（ＰＭＩＢ）のファイル数（File Number）に０を記入（Ｓ６３６）する。この時点ではパーティション内にはファイルは設定されていない。ファイル設定はファイル登録チケット（ＦＲＴ）を使用して設定可能である。このファイル登録チケット（ＦＲＴ）を使用したファイル登録処理については後述する。

【０５３０】さらに、パーティション管理情報ブロック（ＰＭＩＢ）の空き領域ポインタ（Pointer of Free Area）にパーティション定義ブロック（ＰＤＢ）のスタートポジション（Start Position）をコピーしてパーティションの設定登録を終了する。

【０５３１】次に図６０のステップＳ６２１～Ｓ６２８のパーティション削除処理について説明する。ステップＳ６２１では、パーティション登録チケット（ＰＲＴ）に記述されたパーティションマネージャコード（ＰＭＣ）と同一コードのパーティションがデバイスのメモリ部に存在するか否かを検証する。この判定は、デバイスのメモリ部のパーティション定義ブロック（図１９参照）に受信チケット（ＰＲＴ）の記述コードと同一のコードが記述されているか否かを検証することによって判定可能である。

【０５３２】デバイスに同一コード（ＰＭＣ）のパーティションが存在しない場合は、パーティションの削除は不可能であるので、エラー終了とする。同一コードのパーティションがデバイスに存在する場合は、ステップＳ６２２において、削除対象のパーティションより後に生成されたパーティションがデバイスに存在するか否かを判定する。存在しない場合は、削除対象のパーティションが最新のパーティションであり、ステップＳ６２９において削除対象のパーティションのパーティション定義ブロック（ＰＤＢ）（図１９参照）を削除する。

【０５３３】ステップＳ６２２において、削除対象のパーティションより後に生成されたパーティションがデバイスに存在すると判定された場合は、後に生成されたパーティション（後パーティション）のデータを削除対象のパーティションのサイズ（ＰＳ）分、下位にずらす処理を実行（Ｓ６２３）し、さらに、後パーティションのパーティション定義ブロック（ＰＤＢ）を１ブロック上位にずらす処理を実行（Ｓ６２４）する。また、後パーティションのパーティション定義ブロック（ＰＤＢ）に記録されたパーティション開始位置（Partition Start Portion）から削除パーティションのサイズ（ＰＳ）を減算する処理を実行する（Ｓ６２５）。

【０５３４】ステップＳ６２５またはＳ６２９の処理の後、ステップＳ６２６において、デバイス管理情報ブロック（ＤＭＩＢ）（図１５参照）のデバイス（Device）内の空きブロック数（Free Block Number in Device）に削除パーティションのサイズ（ＰＳ）＋１を加算する。＋１は、削除パーティションのパーティション定義ブロック（ＰＤＢ）用のブロックを意味する。

【０５３５】次にステップＳ６２７において、デバイス管理情報ブロック（図１５参照）の空き領域ポインタ（Pointer of Free Area）の値から削除パーティションのサイズ（ＰＳ）を減算する。さらに、ステップＳ６２８において、デバイス管理情報ブロック（図１５参照）のパーティション数（Partition Number）から１を減算、すなわち削除したパーティション数（１）を減算してパーティション登録チケット（ＰＲＴ）に基づくパーティション削除処理が終了する。

【０５３６】以上が、図４７の処理フローにおけるステップＳ４１５のパーティション登録チケット（ＰＲＴ）に基づくパーティション生成、削除処理である。

【０５３７】（パーティション初期登録）次に、図４７の処理フローにおけるステップＳ４０６，Ｓ４１９のパーティション初期データ書き込み処理、すなわちパーティション登録チケット（ＰＲＴ）に基づくパーティション初期登録処理の詳細について図６２以下のフローを用いて説明する。

【０５３８】図６２、図６３、図６４に示す処理フローにおいて、左側がパーティションマネージャの管轄する初期登録装置の処理、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャの管轄する初期登録装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する構成を有する。図４７の処理フローに示すように、図６２の処理開始以前に、初期登録装置とデバイス間では、相互認証が成立し、チケットの正当性、利用者チェックにおいてチケットおよび利用者（チケットユーザであるデバイスアクセス機器としてのリーダライタなど）の正当性が確認され、さらにパーティション登録チケット（ＰＲＴ）に基づくパーティション生成処理が終了しているものとする。また、図６２、図６３、図６４の初期登録装置と、デバイス間のデータの送受信は、相互認証時に生成したセッション鍵Ｋｓｅｓを用いて暗号化されたデータとして送受信される。

【０５３９】図６２のステップＳ６４１において、初期登録装置は、パーティション認証に共通鍵を用いるか否かを判定する。この判定は、使用するパーティション登録チケット（ＰＲＴ）（図２６参照）のAuthentication Type（デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））フィールドを参照して行われる。

【０５４０】図６２に示すように、パーティション認証に共通鍵を用いる場合、ステップＳ６４２～６４３、Ｓ６５１～Ｓ６５４を実行し、パーティション認証に共通鍵を用いない場合、これらのステップは省略される。

【０５４１】パーティション認証に共通鍵を用いる場合、ステップＳ６４２において初期登録装置は、共通鍵

認証データ書き込みコマンドとして、MKauth_PAR_A：双方向個別鍵認証用マスター鍵、Kauth_PAR_B：双方向個別鍵認証用共通鍵、IRL_PAR:排除デバイス（Device）のデバイス識別子（ＩＤ）を登録したリボケーションリスト（Revocation List（Device ID））、およびこれらのバージョン情報をデバイスに送信する。

【０５４２】ステップＳ６５１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ６５２において、受領データをパーティション鍵領域（図２３参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ６５３）し、書き込み終了通知を登録装置に送信（Ｓ６５４）する。

【０５４３】書き込み終了通知を受信（Ｓ６４３）した登録装置は、ステップＳ６４４においてパーティション認証に公開鍵を用いるか否かを判定する。図６２に示すように、パーティション認証に公開鍵を用いる場合、ステップＳ６４５～６４９、Ｓ６５５～Ｓ６６２を実行し、パーティション認証に公開鍵を用いない場合、これらのステップは省略される。

【０５４４】パーティション認証に公開鍵を用いる場合、ステップＳ６４５において登録装置は、公開鍵認証データ書き込みコマンドとして、PUB_CA(PAR)：パーティションマネージャ対応公開鍵証明書を発行する認証局ＣＡ（ＰＡＲ）の公開鍵、PARAM_PAR：パーティション（Partition）の公開鍵パラメータ、CRL_PAR：排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List（Certificate））、およびこれらのバージョン情報をデバイスに送信する。

【０５４５】ステップＳ６５５でデバイスは、上述の書き込みコマンドを受信し、ステップＳ６５６において、受領データをパーティション鍵領域（図２３参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ６５７）し、書き込み終了通知を登録装置に送信（Ｓ６５８）する。

【０５４６】書き込み終了通知を受信（Ｓ６４６）した登録装置は、公開鍵と秘密鍵の鍵ペア生成コマンドをデバイスに送信（Ｓ６４７）する。なお、この実施例では、鍵ペアの生成はデバイスが実行する構成としているが、例えば登録装置が実行してデバイスに提供する構成としてもよい。

【０５４７】鍵ペア生成コマンドを受信（Ｓ６５９）したデバイスは、デバイス内の暗号処理部（図５参照）において公開鍵（ＰＵＢ ＰＡＲ）と秘密鍵（ＰＲＩ ＰＡＲ）のペアを生成し、生成した鍵をパーティション鍵領域（図２３参照）に書き込む（Ｓ６６０）。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ６６１）し、生成格納した公開鍵を登録装置に送信（Ｓ６６２）する。

【０５４８】登録装置は、デバイスから公開鍵（ＰＵＢ ＰＡＲ）を受信（Ｓ６４８）し、先にデバイスから受信したデバイスの識別子ＩＤmとともに、パーティションマネージャ内のデータベース（ＤＢ（ＰＡＲ））（図９参照））に保存する。

【０５４９】次に、パーティションマネージャの登録装置は、ファイル登録チケット（ＦＲＴ：File Registration Ticket）の検証処理に共通鍵を用いるか否かを判定（Ｓ６７１）する。チケット検証には、前述したようにＭＡＣ値検証等による共通鍵方式と、秘密鍵による署名生成、公開鍵による署名検証を行なう公開鍵方式のいずれかを適用することが可能であり、パーティションマネージャは、デバイスの採用する検証処理方式を設定することができる。パーティションマネージャは、デバイスの採用するＦＲＴチケット検証方式に応じて共通鍵、公開鍵のいずれか、あるいは両方式を実行可能なデータをデバイスに設定する。

【０５５０】パーティションマネージャは、ファイル登録チケット（ＦＲＴ：File Registration Ticket）の検証処理に共通鍵認証を実行する設定とする場合は、共通鍵方式のＦＲＴ検証に必要な情報（ｅｘ．ＦＲＴ検証共通鍵）をデバイスにセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスに格納しないことになる。

【０５５１】図６３に示すように、ＦＲＴ検証に共通鍵方式を用いる場合、ステップＳ６７２～６７３、Ｓ６８１～Ｓ６８４を実行し、ＦＲＴ検証に共通鍵を用いない場合、これらのステップは省略される。

【０５５２】ＦＲＴ検証に共通鍵を用いる場合、ステップＳ６７２において登録装置は、ＦＲＴ検証共通鍵書き込みコマンドとして、Kfrt：ファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵、およびバージョン情報をデバイスに送信する。

【０５５３】ステップＳ６８１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ６８２において、受領データをパーティション鍵領域（図２３参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ６８３）し、書き込み終了通知を登録装置に送信（Ｓ６８４）する。

【０５５４】書き込み終了通知を受信（Ｓ６７３）した登録装置は、ステップＳ６７４においてＦＲＴ検証に公開鍵を用いるか否かを判定する。図６３に示すように、ＦＲＴ検証に公開鍵を用いる場合、ステップＳ６７５～６７６、Ｓ６８５～Ｓ６９０を実行し、ＦＲＴ検証に公開鍵を用いない場合、これらのステップは省略される。

【０５５５】ＦＲＴ検証に公開鍵を用いる場合、ステップＳ６７５において登録装置は、ＦＲＴ検証データ書き込みコマンドとして、FRTIC（FRT Issuer Category）：

ファイル登録チケット（ＦＲＴ）発行者カテゴリ、PUB_CA(PAR)：パーティションマネージャ対応公開鍵証明書を発行する認証局ＣＡ（ＰＡＲ）の公開鍵、PARAM_PAR：パーティション（Partition）の公開鍵パラメータ、CRL_PAR：排除デバイス（Device）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)）、およびこれらのバージョン情報をデバイスに送信する。

【０５５６】ステップＳ６８５でデバイスは、上述の書き込みコマンドを受信し、ステップＳ６８６において、受領データ中のFRTIC（FRT Issuer Category）：ファイル登録チケット（ＦＲＴ）発行者カテゴリを公開鍵系パーティション鍵定義ブロック（ＰＫＤＢ：Partition Key Definition block（PUB）（図２２参照））に書き込みバージョン情報を同ブロックのバージョン領域に書き込む。

【０５５７】次にデバイスは、ステップＳ６８７において、PUB_CA(PAR)：パーティションマネージャ対応公開鍵証明書を発行する認証局ＣＡ（ＰＡＲ）の公開鍵データが書き込み済みか否かを判定し、書き込まれていない場合にステップＳ６８８において、PUB_CA(PAR)、PARAM_PAR、CRL_PARをパーティション鍵領域（図２３参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ６８９）し、書き込み終了通知を登録装置に送信（Ｓ６９０）する。

【０５５８】書き込み終了通知を受信（Ｓ６７６）した登録装置は、次に、ステップＳ７０１において、共通鍵データの更新をサポートするデバイスとするか否かを判定する。デバイスに格納されたデータ中、そのいくつかは更新対象データとして前述したデータアップデートチケット（ＤＵＴ：Data Update Ticket）（図３２参照）を用いて更新が可能である。更新対象となるデータは、先に図３３を用いて説明した通りである。このデータアップデートチケット（ＤＵＴ：Data Update Ticket）を用いた更新処理においても共通鍵方式、または公開鍵方式のいずれかの方式が可能であり、パーティションマネージャは設定したパーティションに応じていずれかの方式または両方式を実行可能なデータをデバイスを設定する。

【０５５９】パーティションマネージャは、設定したパーティションを共通鍵方式によるデータ更新を実行する構成とする場合は、共通鍵方式のデータ更新処理に必要な情報（ｅｘ．データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵他）をデバイスのパーティション鍵領域にセットし、デバイスが共通鍵認証を実行しないデバイスであれば、これらの情報をデバイスのパーティション鍵領域に格納しない処理をする。

【０５６０】図６４に示すように、データアップデートチケット（ＤＵＴ：Data Update Ticket）を用いたデータ更新処理に共通鍵方式を用いる場合、ステップＳ７０２～７０３、Ｓ７１１～Ｓ７１４を実行し、データ更新に共通鍵方式を用いない場合、これらのステップは省略される。

【０５６１】データ更新に共通鍵を用いる場合、ステップＳ７０２において登録装置は、データアップデートチケット（ＤＵＴ：Data Update Ticket）検証共通鍵書き込みコマンドとして、Kdut_PAR1：データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵、Kdut_PAR2：データ更新用暗号鍵、Kdut_PAR3：データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵、Kdut_PAR4：データ更新用暗号鍵およびこれらのバージョン情報をデバイスに送信する。

【０５６２】ステップＳ７１１でデバイスは、上述の書き込みコマンドを受信し、ステップＳ７１２において、受領データをパーティション鍵領域（図２３参照）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ７１３）し、書き込み終了通知を登録装置に送信（Ｓ７１４）する。

【０５６３】書き込み終了通知を受信（Ｓ７０３）した登録装置は、ステップＳ７０４において、デバイスに設定したパーティションが公開鍵方式を用いたデータアップデートチケット（ＤＵＴ：Data Update Ticket）を使用したデータ更新処理をサポートするか否かを判定する。図６４に示すように、公開鍵方式をサポートする場合、ステップＳ７０５～７０６、Ｓ７１５～Ｓ７１８を実行し、公開鍵方式をサポートしない場合、これらのステップは省略される。

【０５６４】公開鍵方式をサポートする場合、ステップＳ７０５において登録装置は、データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者コード書き込みコマンドとして、DUTIC_PAR（DUT Issuer Category）：データアップデートチケット（ＤＵＴ：Data Update Ticket）発行者カテゴリ、およびバージョン情報をデバイスに送信する。

【０５６５】ステップＳ７１５でデバイスは、上述の書き込みコマンドを受信し、ステップＳ７１６において、受領データを公開鍵系パーティション鍵定義ブロック（ＰＫＤＢ（ＰＵＢ）：Partition Key Definition Block(PUB)）に書き込む。次にデータ書き込みによって生じたポインタ、サイズ、デバイス内のフリーブロック数の調整を実行（Ｓ７１７）し、書き込み終了通知を登録装置に送信（Ｓ７１８）し、登録装置が書き込み終了通知を受信（Ｓ７０６）して処理を終了する。

【０５６６】パーティションマネージャによる初期登録処理（図６２～図６４の処理フロー）が完了した状態のデバイスのメモリ内格納データ構成例を図６５に示す。図６５に示すパーティション(Partition)領域中、パーティション鍵領域には、上記のフロー（図６２～図６

4）において、登録装置から送信されて下記のデータが書き込まれる。

* IRL_PAR :パーティションアクセス排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の識別子（ＩＤ）を登録したリボケーションリスト（Revocation List (Device ID)）
* CRL_PAR :パーティションアクセス排除デバイス（Device）、排除機器（デバイスアクセス機器としてのリーダライタ、ＰＣ等のチケットユーザ、チケット発行手段）の公開鍵証明書識別子（ｅｘ．シリアルナンバ：ＳＮ）を登録したリボケーションリスト（Revocation List (Certificate)
* Kauth_PAR_B :双方向個別鍵認証用共通鍵
* Mkauth_PAR_A :双方向個別鍵認証用マスター鍵
* Kdut_PAR1 :データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_PAR2 :データ更新用暗号鍵
* Kdut_PAR3 : データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵
* Kdut_PAR4 :データ更新用暗号鍵
* Kfrt :ファイル登録チケット（ＦＲＴ）のＭＡＣ検証用鍵

【０５６７】また、

* PUB_PAR :パーティション（Partition）の公開鍵
* PRI_PAR :パーティション（Partition）の秘密鍵

が、デバイスにおいて生成されて書き込まれる。

【０５６８】また、

* PARAM_PAR :パーティション（Partition）の公開鍵パラメータ
* PUB_CA(PAR) :認証局ＣＡ（ＰＡＲ）の公開鍵

共通鍵系パーティション鍵情報ブロック（Partition Key Definition Block(Common)）
公開鍵系パーティション鍵情報ブロック（Partition Key Definition Block(PUB)）
パーティション管理情報ブロック（Partition Management Information Block）
の各データは、パーティションの生成時（処理フロー図６０、図６１参照）に書き込まれるデータである。

【０５６９】［Ｂ４．２．パーティションマネージャ管理下における公開鍵証明書発行処理］次に図６６以下を用いて、パーティションマネージャによるパーティション対応公開鍵証明書の発行処理について説明する。デバイスには、デバイス全体の認証、デバイスを単位とした処理に適用可能なデバイス対応公開鍵証明書（ＣＥＲＴ ＤＥＶ）と、デバイス内の特定のパーティションに対する処理の際の認証その他検証処理等に適用可能なパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）が格納され得る。パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）は、デバイスに設定されたパーティション毎に設定格納可能である。

【０５７０】パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）は、パーティションマネージャの管轄する登録局を介して認証局（ＣＡ ｆｏｒ ＰＭ）（図２、図３参照）の発行した公開鍵証明書をデバイスに付与する手続きにより発行され、登録局はパーティションマネージャの管轄登録局が発行した公開鍵証明書（ＣＥＲＴＰＡＲ）についての管理（データベース３３２（図９参照））を実行する。

【０５７１】図６６および図６７に従って、パーティションマネージャの管轄登録局による設定パーティションに対するパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の発行処理の手順を説明する。図６６，図６７において、左側がパーティションマネージャの管轄登録局のＣＥＲＴ（公開鍵証明書）発行装置、具体的には、図９に示すパーティションマネージャの構成図における制御手段３３１の処理、右側がデバイスの処理である。

【０５７２】まずステップＳ７２１において、ＣＥＲＴ発行装置は、パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の発行対象となるデバイスのユーザ情報を取得し、証明書発行の許可（判定）を行ない発行対象となるデバイスとの通信路を確保する。パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の発行対象となるデバイスのユーザ情報は、例えばデバイスの初期登録時に生成したデータから取得可能である。なお、ユーザ情報はデバイスとの通信路設定後、デバイスから取得してもよい。通信路は、有線、無線を問わずデータ送受信可能な通信路として確保されればよい。

【０５７３】次にＣＥＲＴ発行装置は、ステップＳ７２２において、乱数Ｒを含む認証データ生成コマンドをデバイスに対して送信する。認証データ生成コマンドを受信（Ｓ７３１）したデバイスは、受信乱数Ｒと、デバイス識別子（ＩＤｍ）の結合データにデバイス秘密鍵（ＰＲＩ ＰＡＲ）を適用してデジタル署名（Ｓ）の生成処理（図１２参照）を実行（Ｓ７３２）する。デバイスは、デバイスの識別データ（ＩＤｍ）と署名（Ｓ）をＣＥＲＴ発行装置に送信する。

【０５７４】デバイスから識別データ（ＩＤｍ）と署名（Ｓ）を受信（Ｓ７２３）したＣＥＲＴ発行装置は、受信したデバイス識別データ（ＩＤｍ）を検索キーとしてデータベースＤＢ（ＰＡＲ）３３２から格納済みのデバイス公開鍵（ＰＵＢ ＰＡＲ）を取得する。さらに、取得したデバイス公開鍵（ＰＵＢ ＰＡＲ）を適用して署名（Ｓ）の検証処理（図１３参照）を実行（Ｓ７２５）する。検証に成功しなかった場合は、デバイスからの送信データは不正なデータであると判定し処理は終了する。

【０５７５】検証に成功した場合は、認証局（ＣＡ ｆｏｒ ＰＭ）６２０に対してパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の発行処理を依頼（Ｓ７２

7）する。パーティションマネージャは認証局６２０の発行したパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）を受信（Ｓ７２８）してデバイスに送信（Ｓ７２９）する。

【０５７６】パーティションマネージャ（登録局）からパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）を受信したデバイスは、予めパーティション鍵領域（図２３参照）に格納済みの認証局の公開鍵（ＰＵＢ ＣＡ（ＰＡＲ））を用いて受信したパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の署名検証を実行する。すなわち公開鍵証明書には認証局の秘密鍵で実行され署名があり（図１１参照）、この署名検証（Ｓ７３５）を行なう。

【０５７７】署名検証に失敗した場合は、正当な公開鍵証明書でないと判定し、エラー通知をＣＥＲＴ発行装置に対して実行（Ｓ７４５）する。

【０５７８】署名検証に成功した場合は、パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）に格納されたデバイス公開鍵（ＰＵＢ ＰＡＲ）と自デバイスに保管されたデバイス公開鍵（ＰＵＢ ＰＡＲ）の比較を実行（Ｓ７４１）し、一致しない場合はエラー通知を実行し、一致した場合は、受信したパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）をパーティション鍵領域（図２３参照）に格納（Ｓ７４３）する。なお、パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の発行以前は、この領域に自デバイスで生成した公開鍵（ＰＵＢ ＰＡＲ）を格納し、正当なパーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）が発行された時点で、パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）により上書きする処理として格納する。

【０５７９】パーティション対応公開鍵証明書（ＣＥＲＴ ＰＡＲ）の格納が終了すると格納処理終了通知をＣＥＲＴ発行装置に送信（Ｓ７４４）する。ＣＥＲＴ発行装置は、格納処理終了通知を受信（Ｓ７５１）し、格納成功を確認（Ｓ７５２）して処理を終了する。格納成功の確認が得られない場合はエラーとして処理が終了する。

【０５８０】［Ｂ４．３．パーティション生成処理各方式における処理手順］上述したように、パーティションの設定登録処理において、パーティションマネージャの管理するデバイスアクセス機器としてのリーダライタとデバイス間において、相互認証が実行され、パーティション登録チケット（ＰＲＴ）に基づくパーティションの設定がなされる。上述したように相互認証処理の態様は、公開鍵相互認証、共通鍵相互認証の２種類のいずれかであり、またチケット（ＰＲＴ）の検証処理も公開鍵系の署名検証、共通鍵系のＭＡＣ検証の２種類のいずれかが実行されることになる。すなわち処理態様としては大きく分けて、

（Ａ）相互認証（公開鍵）、チケット（ＰＲＴ）検証（公開鍵）

（Ｂ）相互認証（公開鍵）、チケット（ＰＲＴ）検証（共通鍵）

（Ｃ）相互認証（共通鍵）、チケット（ＰＲＴ）検証（共通鍵）

（Ｄ）相互認証（共通鍵）、チケット（ＰＲＴ）検証（公開鍵）

の４態様がある。

【０５８１】これらの４態様についての処理を、認証局（ＣＡ（ＤＭ））、デバイスマネージャ（ＤＭ）、パーティションマネージャ（ＰＭ）、デバイス、各エンティティ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【０５８２】（Ａ）相互認証（公開鍵）、チケット（ＰＲＴ）検証（公開鍵）

まず、相互認証処理に公開鍵方式を適用し、チケット（ＰＲＴ）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図６８を用いて説明する。なお以下では、説明を簡略化するために図６８に示すように、認証局（ＣＡ）を１つとし、登録局をデバイスマネージャ内に１つ設定し、デバイスマネージャ公開鍵証明書（Ｃｅｒｔ．ＤＭ）、パーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）の双方をこれらの各登録局、認証局を介して発行する構成とした。またパーティション登録チケット（ＰＲＴ）の発行手段はデバイスマネージャ（ＤＭ）であり、パーティション登録チケット（ＰＲＴ）に対する署名はデバイスマネージャの秘密鍵を用いて実行される。

【０５８３】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

（１）デバイスマネージャ（ＤＭ）の公開鍵証明書（Ｃｅｒｔ．ＤＭ）の発行、公開鍵証明書（Ｃｅｒｔ．ＤＭ）は、認証局（ＣＡ）によってデバイスマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。

（２）パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）の発行、公開鍵証明書（Ｃｅｒｔ．ＰＭ）は、認証局（ＣＡ）によってパーティションマネージャの発行要求により、登録局を介した証明書発行手続きによってパーティションマネージャに対して発行される。

【０５８４】（３）パーティション登録チケット（ＰＲＴ）の発行処理

パーティション登録チケット（ＰＲＴ）は、デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴ Ticket Issuer）によりパーティションマネージャ（ＰＭ）に対して発行される。この場合、公開鍵方式の署名生成、検証を実行するため、デバイスマネージャの秘密鍵による署名(Signature)が生成（図１２参

照）されてＰＲＴに付加される。
（４）ＰＲＴおよびDM公開鍵証明書（Ｃｅｒｔ．ＤＭ）のＰＭに対する供給デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴTicket Issuer）により発行されたパーティション登録チケット（ＰＲＴ）は、DM公開鍵証明書（Ｃｅｒｔ．DM）とともにパーティションマネージャに対して送信される。
【０５８５】（５）ＰＭとデバイス間の相互認証
発行されたＰＲＴに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ（具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ）は、公開鍵方式の相互認証（図５０参照）を実行する。
【０５８６】（６）ＰＲＴおよびDM公開鍵証明書（Ｃｅｒｔ．DM）のデバイスに対する供給
ＰＭとデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）は、デバイスに対してパーティション登録チケット（ＰＲＴ）、およびDM公開鍵証明書（Ｃｅｒｔ．DM）を送信する。デバイスは、受信したパーティション登録チケット（ＰＲＴ）について、（１）チケット発行者（Ticket Issuer）＝DMの公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＤＫＤＢ（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＰＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＰＲＴチケットに格納されたＰＲＴユーザ（この場合はＰＭ：チケットユーザであるデバイスアクセス機器としてのリーダライタ）と受信したパーティションマネージャの公開鍵証明書の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）の一致を確認し、相互認証済みであることを確認することによりＰＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。
【０５８７】（７）パーティションの生成
パーティション登録チケット（ＰＲＴ）の検証、ＰＲＴ発行者（PRT Issuer）、ＰＲＴユーザの検証に成功すると、パーティション登録チケット（ＰＲＴ）に記述されたルールに従ってパーティションがデバイスのメモリ部に生成（図６０、図６１参照）される。
【０５８８】（８）鍵データ書き込み
パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。（９）公開鍵の読み出し、
（１０）公開鍵証明書の発行
生成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域（図２３参照）に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この（９）、（１０）の処理は、作成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）の際に公開鍵認証を行なう構成の場合に実行すればよい。
【０５８９】以上の処理によって、相互認証（公開鍵）、チケット（ＰＲＴ）検証（公開鍵）の各方式に従ったパーティションの生成処理が実行される。
【０５９０】（Ｂ）相互認証（公開鍵）、チケット（ＰＲＴ）検証（共通鍵）
次に、相互認証処理に公開鍵方式を適用し、チケット（ＰＲＴ）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図６９を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。
【０５９１】（１）パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）の発行、公開鍵証明書（Ｃｅｒｔ．ＰＭ）は、認証局（ＣＡ）によってパーティションマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。
【０５９２】（２）パーティション登録チケット（ＰＲＴ）の発行処理
パーティション登録チケット（ＰＲＴ）は、デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴ Ticket Issuer）によりパーティションマネージャ（ＰＭ）に対して発行される。この場合、共通鍵方式の検証値としてＭＡＣ(Message Authentication Code)（図５９参照）がＰＲＴに付加される。
（３）ＰＲＴのＰＭに対する供給
デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴTicket Issuer）により発行されたパーティション登録チケット（ＰＲＴ）は、パーティションマネージャに対して送信される。
【０５９３】（４）ＰＭとデバイス間の相互認証
発行されたＰＲＴに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ（具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ）は、公開鍵方式の相互認証（図５０参照）を実行する。

【０５９４】（５）ＰＲＴの送信
パーティションマネージャは発行されたパーティション登録チケット（ＰＲＴ）をデバイスに送付する。デバイスは、受信したパーティション登録チケット（ＰＲＴ）についてＭＡＣ検証処理を実行し、ＰＲＴ発行者（PRT Issuer）の検証、さらに、ＰＲＴチケットに格納されたＰＲＴユーザ（この場合はＰＭ：チケットユーザであるデバイスアクセス機器としてのリーダライタ）と受信したパーティションマネージャの公開鍵証明書の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）の一致を確認し相互認証済みであることを確認することによりＰＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。
【０５９５】（６）パーティションの生成
パーティション登録チケット（ＰＲＴ）の検証、ＰＲＴ発行者（PRT Issuer）、ＰＲＴユーザの検証に成功すると、パーティション登録チケット（ＰＲＴ）に記述されたルールに従ってパーティションがデバイスのメモリ部に生成（図６０、図６１参照）される。
（７）鍵データ書き込み
パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。
【０５９６】（８）公開鍵の読み出し、
（９）公開鍵証明書の発行
生成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域（図２３参照）に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この（８）、（９）の処理は、作成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）の際に公開鍵認証を行なう構成の場合に実行すればよい。
【０５９７】以上の処理によって、相互認証（公開鍵）、チケット（ＰＲＴ）検証（共通鍵）の各方式に従ったパーティションの生成処理が実行される。
【０５９８】（Ｃ）相互認証（共通鍵）、チケット（ＰＲＴ）検証（共通鍵）
次に、相互認証処理に共通鍵方式を適用し、チケット（ＰＲＴ）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図７０を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。
【０５９９】（１）パーティション登録チケット（ＰＲＴ）の発行処理
パーティション登録チケット（ＰＲＴ）は、デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴ Ticket Issuer）によりパーティションマネージャ（ＰＭ）に対して発行される。この場合、共通鍵方式の検証値としてＭＡＣ（図５９参照）がＰＲＴに付加される。
【０６００】（２）ＰＲＴのＰＭに対する供給
デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴTicket Issuer）により発行されたパーティション登録チケット（ＰＲＴ）は、パーティションマネージャに対して送信される。
【０６０１】（３）ＰＭとデバイス間の相互認証
発行されたＰＲＴに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ（具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ）は、共通鍵方式の相互認証（図５３、図５４参照）を実行する。
【０６０２】（４）ＰＲＴの送信
パーティションマネージャは発行されたパーティション登録チケット（ＰＲＴ）をデバイスに送付する。デバイスは、受信したパーティション登録チケット（ＰＲＴ）についてＭＡＣ検証処理を実行し、ＰＲＴ発行者（PRT Issuer）の検証、さらに、ＰＲＴチケットに格納されたＰＲＴユーザ（この場合はＰＭ：チケットユーザであるデバイスアクセス機器としてのリーダライタ）とパーティションマネージャの識別子の一致を確認し、相互認証済みであることを確認することによりＰＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。
【０６０３】（５）パーティションの生成
パーティション登録チケット（ＰＲＴ）の検証、ＰＲＴ発行者（PRT Issuer）、ＰＲＴユーザの検証に成功すると、パーティション登録チケット（ＰＲＴ）に記述されたルールに従ってパーティションがデバイスのメモリ部に生成（図６０、図６１参照）される。
【０６０４】（６）鍵データ書き込み
パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。
（７）公開鍵の読み出し、
（８）公開鍵証明書の発行
生成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して

公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域（図２３参照）に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この（７）、（８）の処理は、作成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）の際に公開鍵認証を行なう構成の場合に実行すればよい。

【０６０５】以上の処理によって、相互認証（共通鍵）、チケット（ＰＲＴ）検証（公開鍵）の各方式に従ったパーティションの生成処理が実行される。

【０６０６】（Ｄ）相互認証（共通鍵）、チケット（ＰＲＴ）検証（公開鍵）

次に、相互認証処理に共通鍵方式を適用し、チケット（ＰＲＴ）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図７１を用いて説明する。図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【０６０７】（１）デバイスマネージャ（ＤＭ）の公開鍵証明書（Ｃｅｒｔ．ＤＭ）の発行、公開鍵証明書（Ｃｅｒｔ．ＤＭ）は、認証局（ＣＡ）によってデバイスマネージャの発行要求により、登録局を介した証明書発行手続きによってデバイスマネージャに対して発行される。

【０６０８】（２）パーティション登録チケット（ＰＲＴ）の発行処理

パーティション登録チケット（ＰＲＴ）は、デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴ Ticket Issuer）によりパーティションマネージャ（ＰＭ）に対して発行される。この場合、公開鍵方式の署名生成、検証を実行するため、デバイスマネージャの秘密鍵による署名(Signature)が生成（図１２参照）されてＰＲＴに付加される。

（３）ＰＲＴおよびＤＭ公開鍵証明書（Ｃｅｒｔ．ＤＭ）のＰＭに対する供給デバイスマネージャの管理するパーティション登録チケット発行手段（ＰＲＴTicket Issuer）により発行されたパーティション登録チケット（ＰＲＴ）は、ＤＭ公開鍵証明書（Ｃｅｒｔ．ＤＭ）とともにパーティションマネージャに対して送信される。

【０６０９】（４）ＰＭとデバイス間の相互認証

発行されたＰＲＴに従ったパーティションを生成しようとする対象のデバイスと、パーティションマネージャ（具体的にはチケットユーザであるデバイスアクセス機器としてのリーダライタ）は、共通鍵方式の相互認証（図５３、図５４参照）を実行する。

【０６１０】（５）ＰＲＴおよびＤＭ公開鍵証明書（Ｃｅｒｔ．ＤＭ）のデバイスに対する供給

ＰＭとデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）は、デバイスに対してパーティション登録チケット（ＰＲＴ）、およびＤＭ公開鍵証明書（Ｃｅｒｔ．ＤＭ）を送信する。デバイスは、受信したパーティション登録チケット（ＰＲＴ）について、（１）チケット発行者（Ticket Issuer）＝ＤＭの公開鍵証明書（ＣＥＲＴ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ）のオプション領域に記録されたコードと、デバイス内のＤＫＤＢ（Device Key Definition Block）(PUB)に記録されたチケット発行手段コード（ＰＲＴＩＣ：PRT Issuer Code）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＰＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＰＲＴチケットに格納されたＰＲＴユーザ（この場合はＰＭ：チケットユーザであるデバイスアクセス機器としてのリーダライタ）とパーティションマネージャの公開鍵証明書中の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）の一致を確認し、相互認証済みであることを確認することによりＰＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０６１１】（６）パーティションの生成

パーティション登録チケット（ＰＲＴ）の検証、ＰＲＴ発行者（PRT Issuer）、ＰＲＴユーザの検証に成功すると、パーティション登録チケット（ＰＲＴ）に記述されたルールに従ってパーティションがデバイスのメモリ部に生成（図６０、図６１参照）される。

【０６１２】（７）鍵データ書き込み

パーティションがデバイスのメモリ部に生成されると、生成されたパーティション内に対する各種鍵の格納処理が実行される。

（８）公開鍵の読み出し、

（９）公開鍵証明書の発行

生成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）に際し、公開鍵認証を行なう場合、デバイスは公開鍵、秘密鍵の鍵ペアを生成し、生成した公開鍵をパーティションマネージャに送信し、登録局、認証局を介して公開鍵証明書の発行処理を行ない、発行された公開鍵証明書をパーティション鍵領域（図２３参照）に格納する。この際、生成した公開鍵の格納領域に対して発行された公開鍵証明書を格納する。なお、この（８）、（９）の処理は、作成パーティションに対する各種サービス時の認証処理（パーティション生成、ファイル生成、ファイルアクセス、データアップデート等のサービス利用時の認証処理）の際に公開鍵認証を行なう構成の場合に実行すればよい。

【０６１３】以上の処理によって、相互認証（共通

鍵）、チケット（ＰＲＴ）検証（公開鍵）の各方式に従ったパーティションの生成処理が実行される。

【０６１４】［Ｂ４．４．ファイル登録チケット（ＦＲＴ）を利用したファイル生成、削除処理］次に、デバイスに生成したパーティション内にファイル登録チケット（ＦＲＴ）を適用してファイルを生成、または削除する処理について説明する。図７２以下のフロー他の図面を参照して説明する。なお、ファイル作成、削除処理には、デバイスとデバイスアクセス機器としてのリーダライタ（パーティションマネージャ）間における相互認証処理（デバイス認証またはパーティション認証）、パーティション登録チケット（ＦＲＴ：File Registration Ticket）の正当性検証処理が含まれる。

【０６１５】図７２に示すファイル生成、削除処理フローについて説明する。図７２において、左側がパーティションマネージャのファイル作成・削除装置、右側がデバイス（図５参照）の処理を示す。なお、パーティションマネージャのファイル作成・削除装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する。まず、図７２を用いて、ファイル作成、削除処理の概要を説明し、その後、当処理に含まれるファイル作成、削除操作の詳細を図７３のフローを用いて説明する。

【０６１６】まず、図７２のステップＳ８０１とＳ８１０において、ファイル作成・削除装置とデバイス間にでの相互認証処理が実行される。データ送受信を実行する２つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【０６１７】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、パーティション認証が実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図４８～図５６を用いて説明したと同様の処理であるので説明を省略する。

【０６１８】なお、相互認証処理として実行すべき処理は、適用するファイル登録チケット（ＦＲＴ）（図２７参照）の

* Authentication Flag ：チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

* Authentication Type ：デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））によって決定される。

【０６１９】認証処理に失敗した場合（Ｓ８０２，Ｓ８１１でＮｏ）は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【０６２０】認証処理に成功すると、ファイル作成・削除装置は、デバイスに対してファイル登録チケット（ＦＲＴ：File Registration Ticket）を送信する。ファイル登録チケット（ＦＲＴ）は、パーティションマネージャの管理下のファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）により発行されるチケットである。ファイル登録チケット（ＦＲＴ）は、デバイスに対するアクセス制御チケットであり、先に説明した図２７のデータフォーマット構成を持つチケットである。

【０６２１】なお、ファイル登録チケット（ＦＲＴ）を、チケットユーザに対して送信する際には、公開鍵方式の場合、ファイル登録チケット（ＦＲＴ）発行手段（FRT Issuer）の公開鍵証明書（CERT_FRTI）も一緒に送信する。ＦＲＴ発行手段の公開鍵証明書（CERT_FRTI）の属性（Attribute）は、、ファイル登録チケット（ＦＲＴ）発行手段（ＦＲＴ Issuer）の識別子(FRTIC)と一致する。

【０６２２】ファイル登録チケット（ＦＲＴ）を受信（Ｓ８１２）したデバイスは、受信したチケット（ＦＲＴ）の正当性と利用者チェック処理を実行（Ｓ８１３）する。チケットの正当性の検証処理は、共通鍵方式によるＭＡＣ検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器（チケット利用者）の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子（図２７参照）との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット（ＰＲＴ）の適用処理についての説明中、図５７～図５９を用いて説明したと同様の処理であるので説明を省略する。

【０６２３】デバイスにおいて、受信チケット（ＦＲＴ）の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合（Ｓ８１４でＮｏ）は、ファイル登録チケット（ＦＲＴ）受理エラーをファイル作成・削除装置に通知（Ｓ８１８）する。チケットおよび利用者の正当なことが確認できた場合（Ｓ８１４でＹｅｓ）は、受信したファイル登録チケット（ＦＲＴ）に記述されたルールに従いデバイス内のメモリ部におけるファイルの生成、または削除処理を実行する。この処理の詳細については、別フローを用いて後段で詳述する。

【０６２４】ファイル登録チケット（ＦＲＴ）の記述に従って、ファイルの生成または削除処理に成功（Ｓ８１６でＹｅｓ）すると、ＦＲＴ受理成功をファイル作成・

削除装置に通知（Ｓ８１７）する。一方、ファイルの生成または削除処理に失敗（Ｓ８１６でＮｏ）した場合は、ＦＲＴ受理エラーをファイル作成・削除装置に通知（Ｓ８１８）する。

【０６２５】ファイル作成・削除装置は、ＦＲＴ受理結果を受信（Ｓ８０４）し、ＦＲＴ処理結果を判定し、ＦＲＴ受理結果がエラーである場合（Ｓ８０５でＮｏ）は、エラーとして処理を終了し、ＦＲＴ受理結果が成功（Ｓ８０５でＹｅｓ）である場合はセッションクリアコマンドの送受信（Ｓ８０６，Ｓ８１９）を実行し、デバイス側に生成した認証テーブルを破棄（Ｓ８２０）し、処理を終了する。認証テーブルは、ステップＳ８０１，Ｓ８１０の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット（ＰＲＴ）の適用処理の項目において説明した構成、すなわち、図５１の構成と同様のものである。

【０６２６】このようにファイル登録チケット（ＦＲＴ）を利用して、デバイス内に設定されたパーティション内にファイルの生成、または生成済みのファイルの削除処理が実行される。以下、当処理に含まれるファイルの生成、削除処理（Ｓ８１５）について、図７３を用いて説明する。

【０６２７】（ファイル作成・削除処理）図７２のフローに示すステップＳ８１５において実行されるファイル登録チケット（ＦＲＴ）に基づくパーティションの作成、削除処理の詳細について、図７３の処理フローを用いて説明する。ファイルの作成、削除処理は、チケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ等）からファイル登録チケット（ＦＲＴ）を受信したデバイスが、ファイル登録チケット（ＦＲＴ）に基づいて実行する処理である。

【０６２８】図７３のステップＳ８２１において、デバイスは、受信したファイル登録チケット（ＦＲＴ:File Registration ticket）に記録された処理タイプ、すなわちOperation Type（パーティション（Partition）作成か削除かの指定（作成（Generate）／削除（Delete））））を検証する。処理タイプ（Operation Type）が、ファイル作成である場合、ステップＳ８２２以下を実行し、ファイル削除である場合、ステップＳ８４１以下を実行する。

【０６２９】まず、ファイル作成処理について説明する。デバイスはステップＳ８２２において、ファイル登録チケット（ＦＲＴ）に記述されたファイル識別子（ＩＤ）と同一ＩＤのファイルがデバイスの処理対象パーティション内に存在するか否かを検証する。この判定は、デバイスのメモリ部に設定されたパーティション領域のファイル定義ブロック（図２４参照）に受信チケット（ＦＲＴ）に記述されたファイルＩＤと同一のファイルＩＤが記述されているか否かを検証することによって判定可能である。

【０６３０】すでにデバイスに同一ＩＤのファイルが存在する場合は、同一ＩＤを持つ重複ファイルを同一のパーティション内に存在させることは許されないので、ファイルの生成は実行せず、エラー終了とする。同一ＩＤのファイルが処理対象パーティション内に存在しない場合は、ステップＳ８２３において、パーティション管理情報ブロック（図２０参照）のパーティション内の空きブロック数（Free Block Number in Partition）と、ファイル登録チケット（ＦＲＴ）に記述されたファイルサイズ（File Size）とを比較し、チケット（ＦＲＴ）に記述されたファイルサイズ（File Size）以上の空きブロック領域がデバイスの処理対象パーティション内に存在するか否かを判定する。存在しない場合は、ＦＲＴに記述されたサイズのファイルの生成はできないので、エラー終了とする。

【０６３１】チケット（ＦＲＴ）に記述されたファイルサイズ（File Size）以上の空きブロック領域がデバイスのメモリ部の処理対象パーティション内に存在すると判定された場合は、ステップＳ８２４に進み、パーティション管理情報ブロックの空き領域ポインタ（Pointer of Free Area）を参照してパーティションの空き領域（Free Area in Partition）の最上位ブロックにファイル定義ブロック（ＦＤＢ）エリア（図２４参照）を確保する。

【０６３２】次に、デバイスは、確保したファイル定義ブロック（ＦＤＢ）エリアに、ファイル登録チケット（ＦＲＴ）に記述されたファイルＩＤのコピーを実行（Ｓ８２５）し、さらに、ファイル定義ブロック（ＦＤＢ）エリアのファイルスタート位置（File Start Position）に、パーティション管理情報ブロック（図２０参照）の空き領域ポインタ（Pointer of Free Area）のコピー処理を実行（Ｓ８２６）する。

【０６３３】さらに、ステップＳ８２７において、ファイル定義ブロック（ＦＤＢ）のファイルサイズ（File Size）、サービス許可チケット発行手段コード（ＳＰＴＩＣ）、およびバージョン、（ＳＰＴＩＣ Version）、ファイル構造タイプ（File Structure Type Code）、ファイルアクセスを行う際に、指定する認証方式（Acceptable Authentication Type）、指定する検証方式（AcceptableVerificztionType）のそれぞれに、ファイル登録チケット（ＦＲＴ）に記述された各対応データをコピーする。

【０６３４】次に、ステップＳ８２８において、ファイル登録チケット（ＦＲＴ）に格納されたKspt_Encrypted（ファイル定義ブロック（File Definition Block）に記載されるサービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵 Kspt を そのパーティションのファイル登録チケットのＭＡＣ検証用鍵 Kfrt で暗号化したデータKfrt（Kspt））をファイル登録チケットのＭＡＣ検証用鍵 Kfrtを用いて復号してファイル定義ブロック（ＦＤＢ）

に格納する。なお、ファイル登録チケットのＭＡＣ検証用鍵 Kfrtは、パーティションの生成時にパーティション鍵領域に格納済みである。

【０６３５】次に、ステップＳ８２９において、パーティション管理情報ブロック（図２０参照）のパーティション（Partition）内の空きブロック数（Free Block Numberin Partition）からファイルサイズ（File Size）＋１を減算する。なお、＋１は、ファイル定義ブロック（ＦＤＢ）用のブロックを意味する。

【０６３６】次に、ステップＳ８３０において、パーティション管理情報ブロック（図２０参照）の空き領域ポインタ（Pointer of Free Area）に生成したファイルサイズ（File Size）を加算し、ステップＳ８３１において、パーティション管理情報ブロックのファイル数（File Number）に１を加算、すなわち生成したファイル数（１）を加算する。

【０６３７】次に、ステップＳ８３２において、ファイル登録チケット（ＦＲＴ）に格納されたFile Structure（生成するファイル（File）のファイル構造（Structure））に応じた初期化処理を実行する。例えばファイル構造がランダム(Random)であれば、０リセット、サイクリック（Cyslic）であれば、ポインタ、データを０リセットするなどの処理を実行する。これらの処理により、生成したパーティション内に新たなファイルが生成される。

【０６３８】次に図７３のステップＳ８４１～Ｓ８４８のファイル削除処理について説明する。ステップＳ８４１では、ファイル登録チケット（ＦＲＴ）に記述されたファイルＩＤと同一ＩＤのファイルがデバイスのメモリ部の処理対象パーティション内に存在するか否かを検証する。この判定は、デバイスのメモリ部のファイル定義ブロック（図２４参照）に受信チケット（ＦＲＴ）に記述されたファイルＩＤと同一のファイルＩＤが記述されているか否かを検証することによって判定可能である。

【０６３９】デバイスの処理対象パーティション内に同一ファイルＩＤのファイルが存在しない場合は、ファイルの削除は不可能であるので、エラー終了とする。同一ＩＤのファイルがデバイスの処理対象パーティション内に存在する場合は、ステップＳ８４２において、削除対象のファイルより後に生成されたファイルが処理対象パーティション内に存在するか否かを判定する。存在しない場合は、削除対象のファイルが最新のファイルであり、ステップＳ８４９において削除対象のファイルのファイル定義ブロック（ＦＤＢ）（図２４参照）を削除する。

【０６４０】ステップＳ８４２において、削除対象のファイルより後に生成されたファイルが処理対象パーティション内に存在すると判定された場合は、後に生成されたファイル（後ファイル）のデータを削除対象のファイルのサイズ（ＦＳ）分、下位にずらす処理を実行（Ｓ８４３）し、さらに、後ファイルのファイル定義ブロック（ＦＤＢ）を１ブロック上位にずらす処理を実行（Ｓ８４４）する。また、後ファイルのファイル定義ブロック（ＦＤＢ）に記録されたファイル開始位置（File Start Portion）から削除ファイルのサイズ（ＦＳ）を減算する処理を実行する（Ｓ８４５）。

【０６４１】ステップＳ８４５またはＳ８４９の処理の後、ステップＳ８４６において、パーティション管理情報ブロック（ＰＭＩＢ）（図２０参照）のパーティション内の空きブロック数（Free Block Number in Partition）に削除ファイルのサイズ（ＦＳ）＋１を加算する。＋１は、削除ファイルのファイル定義ブロック（ＦＤＢ）用のブロックを意味する。

【０６４２】次にステップＳ８４７において、パーティション管理情報ブロック（ＰＭＩＢ）（図２０参照）の空き領域ポインタ（Pointer of Free Area）の値から削除ファイルのサイズ（ＦＳ）を減算する。さらに、ステップＳ８４８において、パーティション管理情報ブロック（ＰＭＩＢ）（図２０参照）のファイル数（File Number）から１を減算、すなわち削除したファイル数（１）を減算してファイル登録チケット（ＦＲＴ）に基づくファイル削除処理が終了する。

【０６４３】以上が、図７２の処理フローにおけるステップＳ８１５のファイル登録チケット（ＦＲＴ）に基づくファイル生成、削除処理である。

【０６４４】パーティションマネージャによるファイル生成処理が完了した状態のデバイスのメモリ内格納データ構成例を図７４に示す。図７４に示すパーティション(Partion)領域中、

ファイル定義ブロック（１～Ｎ）(File Definition Block)

パーティション鍵領域 (Partition Key Area)

共通鍵系パーティション鍵情報ブロック (Partition Key Definition Block(Common))

公開鍵系パーティション鍵情報ブロック (Partition Key Definition Block(PUB))

パーティション管理情報ブロック (Partition Management Information Block)

の各データは、ファイル生成時、またはパーティション生成時に書き込まれるデータである。ファイル領域(File Data Area １～N)は、ファイル生成処理によって処理対象パーティション内にファイル領域として確保される。

【０６４５】［Ｂ４．５．ファイル生成処理各方式における処理手順］上述したファイルの設定登録処理において、パーティションマネージャが管理し、ファイル登録チケットユーザであるデバイスアクセス機器としてのリーダライタとデバイス間において、相互認証が実行され、ファイル登録チケット（ＦＲＴ）に基づくファイルの設定がなされる。相互認証処理の態様は、公開鍵相互

認証、共通鍵相互認証の２種類のいずれかであり、またチケット（ＦＲＴ）の検証処理も公開鍵系の署名検証、共通鍵系のＭＡＣ検証の２種類のいずれかが実行されることになる。すなわち処理態様としては大きく分けて、
（Ａ）相互認証（公開鍵）、チケット（ＦＲＴ）検証（公開鍵）
（Ｂ）相互認証（公開鍵）、チケット（ＦＲＴ）検証（共通鍵）
（Ｃ）相互認証（共通鍵）、チケット（ＦＲＴ）検証（共通鍵）
（Ｄ）相互認証（共通鍵）、チケット（ＦＲＴ）検証（公開鍵）
の４態様がある。

【０６４６】これらの４態様についての処理を、認証局（ＣＡ（ＰＭ））、パーティションマネージャ（ＰＭ）、デバイス、各エンティテイ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【０６４７】（Ａ）相互認証（公開鍵）、チケット（ＦＲＴ）検証（公開鍵）

まず、相互認証処理に公開鍵方式を適用し、チケット（ＦＲＴ）検証に公開鍵方式を適用する場合の各エンティテイ間のデータ転送について図７５を用いて説明する。

【０６４８】図に示す番号順に各エンティテイ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

（１）ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）の発行、ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）は、ファイル登録チケット発行手段（ＦＲＴ Issuer）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））から発行される。なお、パーティションマネージャがファイル登録チケット発行手段（ＦＲＴ Issuer）を兼ねる構成も可能であり、その場合は、ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０６４９】（２）ファイル登録チケットユーザ（ＦＲＴ User）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）の発行、ファイル登録チケットユーザ（ＦＲＴ User：具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）は、ファイル登録チケットユーザ（ＦＲＴ User）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））によって発行される。なお、パーティションマネージャがファイル登録チケットユーザ（ＦＲＴ User）を兼ねる構成も可能であり、その場合は、ファイル登録チケットユーザ（ＦＲＴ User）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０６５０】（３）ファイル登録チケット（ＦＲＴ）の生成処理

ファイル登録チケット（ＦＲＴ）は、パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の秘密鍵による署名(Signature)が生成（図１２参照）されてＦＲＴに付加される。

【０６５１】（４）ＦＲＴおよびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）のファイル登録チケットユーザ（ＦＲＴ User）に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴTicket Issuer）により発行されたファイル登録チケット（ＦＲＴ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）とともにファイル登録チケットユーザ（ＦＲＴ User）すなわち、デバイスに対してチケットを送信する機器（ｅｘ．デバイスアクセス機器としてのリーダライタ）に対して送信される。

【０６５２】（５）ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ（具体的にはファイル登録チケットユーザ（ＦＲＴUser）であるデバイスアクセス機器としてのリーダライタ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ）に従ったファイルを生成しようとする対象のデバイスに対し、チケットユーザ（ＦＲＴ User）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）をデバイスに送信し、公開鍵方式の相互認証（図５０参照）を実行する。

【０６５３】（６）ＦＲＴおよびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）のデバイスに対する供給

パーティションマネージャ（ＰＭ）とデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）（具体的にはファイル登録チケットユーザ（ＦＲＴ User）であるデバイスアクセス機器としてのリーダライタ）は、デバイスに対してファイル登録チケット（ＦＲＴ）、およびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）を送信する。

【０６５４】デバイスは、受信したファイル登録チケット（ＦＲＴ）について、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ FRT Issuer）が改

竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（2）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴFRT Issuer）のオプション領域に記録されたコードと、デバイス内のＰＫＤＢ（Partition Key Definition Block）(PUB)に記録されたチケット発行手段コード（FRTIC：FRT Issuer Category）の一致、（3）チケット発行手段（Ticket Issuer）がリボークされていないこと、（4）受信チケット（ＦＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＦＲＴチケットに格納されたＦＲＴユーザ（チケットユーザであるデバイスアクセス機器としてのリーダライタ）とチケットユーザ（ＦＲＴ User）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）名（ＤＮ）の一致を確認し、相互認証済みであることを確認することによりＦＲＴユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０６５５】（7）ＦＤＢにＳＰＴＩＣおよびＫｓｐｔを登録

デバイスは、ファイル定義ブロック（ＦＤＢ：File Definition Block）にサービス許可チケット（ＳＰＴ）ユーザ（ＳＰＴＩＣ）（ｅｘ．デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ）とKspt（サービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵(Kspt)）を登録する（図７３のフローにおけるステップＳ８２７，Ｓ８２８）。

【０６５６】（8）ファイルデータ領域の確保

デバイスは、処理対象パーティションにファイル登録チケット（ＦＲＴ）に記述されたサイズを持つファイル領域を確保する。

【０６５７】以上の処理によって、相互認証（公開鍵）、チケット（ＦＲＴ）検証（公開鍵）の各方式に従ったファイルの生成処理が実行される。

【０６５８】（Ｂ）相互認証（公開鍵）、チケット（ＦＲＴ）検証（共通鍵）

次に、相互認証処理に公開鍵方式を適用し、チケット（ＦＲＴ）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図７６を用いて説明する。

【０６５９】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【０６６０】（1）ファイル登録チケットユーザ（ＦＲＴ User）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）の発行、ファイル登録チケットユーザ（ＦＲＴ User：具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）は、ファイル登録チケットユーザ（ＦＲＴ User）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））によって発行される。なお、パーティションマネージャがファイル登録チケットユーザ（ＦＲＴ User）を兼ねる構成も可能であり、その場合は、ファイル登録チケットユーザ（ＦＲＴ User）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０６６１】（2）ファイル登録チケット（ＦＲＴ）の生成処理

ファイル登録チケット（ＦＲＴ）は、パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）により生成される。この場合、共通鍵方式の検証値としてＭＡＣ(Message Authentication Code)（図５９参照）がＦＲＴに付加される。

【０６６２】（3）ＦＲＴのファイル登録チケットユーザ（ＦＲＴ User）に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴTicket Issuer）により発行されたファイル登録チケット（ＦＲＴ）は、ファイル登録チケットユーザ（ＦＲＴ User）すなわち、デバイスに対してチケットを送信する機器（ｅｘ．デバイスアクセス機器としてのリーダライタ）に対して送信される。

【０６６３】（4）ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ（具体的にはファイル登録チケットユーザ（ＦＲＴUser）であるデバイスアクセス機器としてのリーダライタ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ）に従ったファイルを生成しようとする対象のデバイスに対し、チケットユーザ（ＦＲＴ User）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ User）をデバイスに送信し、公開鍵方式の相互認証（図５０参照）を実行する。

【０６６４】（5）ＦＲＴのデバイスに対する供給

パーティションマネージャ（ＰＭ）とデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）（具体的にはファイル登録チケットユーザ（ＦＲＴ User）であるデバイスアクセス機器としてのリーダライタ）は、デバイスに対してファイル登録チケット（ＦＲＴ）を送信する。デバイスは、受信したファイル登録チケット（ＦＲＴ）についてＭＡＣ検証処理を実行し、ＦＲＴ発行者（FRT Issuer）の検証、さらに、ＦＲＴチケットに格納されたＦＲＴユーザ（チケットユーザであるデバイスアクセス機器としてのリーダライタ）と受信したパーティションマネージャの公開鍵証明書の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）名（ＤＮ）の一致を確認し相互認証済みであることを確認することによりＦＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０６６５】（６）ＦＤＢにＳＰＴＩＣおよびＫｓｐｔを登録
デバイスは、ファイル定義ブロック（ＦＤＢ：File Definition Block）にサービス許可チケット（ＳＰＴ）発行者カテゴリ（ＳＰＴＩＣ）（ｅｘ．デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ）とKspt（サービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵(Kspt)）を登録する（図７３のフローにおけるステップＳ８２７，Ｓ８２８）。
【０６６６】（８）ファイルデータ領域の確保
デバイスは、処理対象パーティションにファイル登録チケット（ＦＲＴ）に記述されたサイズを持つファイル領域を確保する。
【０６６７】以上の処理によって、相互認証（公開鍵）、チケット（ＦＲＴ）検証（共通鍵）の各方式に従ったファイルの生成処理が実行される。
【０６６８】（Ｃ）相互認証（共通鍵）、チケット（ＦＲＴ）検証（共通鍵）
次に、相互認証処理に共通鍵方式を適用し、チケット（ＦＲＴ）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図７７を用いて説明する。
【０６６９】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。
【０６７０】（１）ファイル登録チケット（ＦＲＴ）の生成処理
ファイル登録チケット（ＦＲＴ）は、パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）により生成される。この場合、共通鍵方式の検証値としてＭＡＣ(Message Authentication Code)（図５９参照）がＦＲＴに付加される。
【０６７１】（２）ＦＲＴのファイル登録チケットユーザ（ＦＲＴ User）に対する供給
パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴTicket Issuer）により発行されたファイル登録チケット（ＦＲＴ）は、ファイル登録チケットユーザ（ＦＲＴ User）すなわち、デバイスに対してチケットを送信する機器（ｅｘ．デバイスアクセス機器としてのリーダライタ）に対して送信される。
【０６７２】（３）ファイル登録チケット発行手段とデバイス間の相互認証
パーティションマネージャ（具体的にはファイル登録チケットユーザ（ＦＲＴUser）であるデバイスアクセス機器としてのリーダライタ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ）に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証（図５３、図５４参照）を実行する。
【０６７３】（４）ＦＲＴのデバイスに対する供給
パーティションマネージャ（ＰＭ）とデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）（具体的にはファイル登録チケットユーザ（ＦＲＴ User）であるデバイスアクセス機器としてのリーダライタ）は、デバイスに対してファイル登録チケット（ＦＲＴ）を送信する。デバイスは、受信したファイル登録チケット（ＦＲＴ）についてＭＡＣ検証処理を実行し、ＦＲＴ発行者（FRT Issuer）の検証、さらに、ＦＲＴチケットに格納されたＦＲＴユーザ（チケットユーザであるデバイスアクセス機器としてのリーダライタ）と受信したパーティションマネージャの識別子の一致を確認し相互認証済みであることを確認することによりＦＲＴユーザ（ＰＭ：デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。
【０６７４】（６）ＦＤＢにＳＰＴＩＣおよびＫｓｐｔを登録
デバイスは、ファイル定義ブロック（ＦＤＢ：File Definition Block）にサービス許可チケット（ＳＰＴ）発行者カテゴリ（ＳＰＴＩＣ）（ｅｘ．デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ）とKspt（サービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵(Kspt)）を登録する（図７３のフローにおけるステップＳ８２７，Ｓ８２８）。
【０６７５】（８）ファイルデータ領域の確保
デバイスは、処理対象パーティションにファイル登録チケット（ＦＲＴ）に記述されたサイズを持つファイル領域を確保する。
【０６７６】以上の処理によって、相互認証（共通鍵）、チケット（ＦＲＴ）検証（共通鍵）の各方式に従ったファイルの生成処理が実行される。
【０６７７】（Ｄ）相互認証（共通鍵）、チケット（ＦＲＴ）検証（公開鍵）
次に、相互認証処理に共通鍵方式を適用し、チケット（ＦＲＴ）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図７８を用いて説明する。
【０６７８】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。
（１）ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）の発行、ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）は、ファイル登録チケット発行手段（ＦＲＴ Issuer）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））から発行される。なお、パーティションマネージャがファイル登録チケット発行手段（ＦＲＴ Issuer）を兼ねる構成も可能であり、その場合は、ファイル登録チケット発行手段（ＦＲＴ Issuer）の公開鍵証明書と

してパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０６７９】（２）ファイル登録チケット（ＦＲＴ）の生成処理

ファイル登録チケット（ＦＲＴ）は、パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の秘密鍵による署名(Signature)が生成（図１２参照）されてＦＲＴに付加される。

【０６８０】（３）ＦＲＴおよびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）のファイル登録チケットユーザ（ＦＲＴ User）に対する供給

パーティションマネージャの管理するファイル登録チケット発行手段（ＦＲＴTicket Issuer）により発行されたファイル登録チケット（ＦＲＴ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）とともにファイル登録チケットユーザ（ＦＲＴ User）すなわち、デバイスに対してチケットを送信する機器（ｅｘ．デバイスアクセス機器としてのリーダライタ）に対して送信される。

【０６８１】（４）ファイル登録チケット発行手段とデバイス間の相互認証

パーティションマネージャ（具体的にはファイル登録チケットユーザ（ＦＲＴUser）であるデバイスアクセス機器としてのリーダライタ）は、ファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）の発行したファイル登録チケット（ＦＲＴ）に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証（図５３、図５４参照）を実行する。

【０６８２】（５）ＦＲＴおよびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）のデバイスに対する供給

パーティションマネージャ（ＰＭ）とデバイス間の相互認証が成立すると、パーティションマネージャ（ＰＭ）（具体的にはファイル登録チケットユーザ（ＦＲＴ User）であるデバイスアクセス機器としてのリーダライタ）は、デバイスに対してファイル登録チケット（ＦＲＴ）、およびファイル登録チケット発行手段（ＦＲＴ Ticket Issuer）公開鍵証明書（Ｃｅｒｔ．ＦＲＴ Issuer）を送信する。

【０６８３】デバイスは、受信したファイル登録チケット（ＦＲＴ）について、（１）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴ FRT Issuer）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴFRT Issuer）のオプション領域に記録されたコードと、、デバイス内のＰＫＤＢ（Partition Key DefinitionBlock）(PUB)に記録されたチケット発行手段コード（FRTIC：FRT Issuer Category）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＦＲＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＦＲＴチケットに格納されたＦＲＴユーザ（チケットユーザであるデバイスアクセス機器としてのリーダライタ）とチケットユーザ（ＦＲＴ User）の識別子の一致を確認し、相互認証済みであることを確認することによりＦＲＴユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０６８４】（６）ＦＤＢにＳＰＴＩＣおよびＫｓｐｔを登録

デバイスは、ファイル定義ブロック（ＦＤＢ：File Definition Block）にサービス許可チケット（ＳＰＴ）発行者カテゴリ（ＳＰＴＩＣ）（ｅｘ．デバイスのファイル内のデータにアクセスを実行するデバイスアクセス機器としてのリーダライタ）とKspt（サービス許可チケット（ＳＰＴ）のＭＡＣ検証用鍵(Kspt)）を登録する（図７３のフローにおけるステップＳ８２７，Ｓ８２８）。

【０６８５】（７）ファイルデータ領域の確保

デバイスは、処理対象パーティションにファイル登録チケット（ＦＲＴ）に記述されたサイズを持つファイル領域を確保する。

【０６８６】以上の処理によって、相互認証（共通鍵）、チケット（ＦＲＴ）検証（公開鍵）の各方式に従ったファイルの生成処理が実行される。

【０６８７】［Ｂ４．６．サービス許可チケット（ＳＰＴ）を利用したサービス（ファイルアクセス）処理］次に、サービス許可チケット（ＳＲＴ）（図２８、図３１参照）を利用したファイルアクセス処理について説明する。図７９以下のフロー他の図面を参照して説明する。なお、ファイルアクセス処理には、デバイスとファイルアクセス装置間における相互認証処理（デバイス認証またはパーティション認証）、サービス許可チケット（ＳＰＴ：Service Parmission Ticket）の正当性検証処理が含まれる。

【０６８８】図７９のフローにおいて、左側がファイルアクセス装置、右側がデバイス（図５参照）の処理を示す。なお、ファイルアクセス装置は、パーティションマネージャの管理装置であり、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する。まず、図７９を用いて、ファイルアクセス装置によるファイルアクセス処理の概要を説明し、その後、当処理に含まれる各処理の詳細を図８０以下のフローを用いて順次説明する。

【０６８９】まず、図７９のステップＳ８５１とＳ８６

０において、ファイルアクセス装置とデバイス間にでの相互認証処理が実行される。データ送受信を実行する２つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【０６９０】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、パーティション認証が実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図４８～図５６を用いて説明したと同様の処理であるので説明を省略する。

【０６９１】なお、相互認証処理として実行すべき処理は、適用するサービス許可チケット（ＳＰＴ）（図２８、図３１参照）の

* Authentication Flag :チケット（Ticket）の利用処理においてデバイス（Device）との相互認証が必要か否かを示すフラグ

* Authentication Type :デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共通鍵認証、または、いずれでも可（Any））

によって決定される。

【０６９２】認証処理に失敗した場合（Ｓ８５２，Ｓ８６１でＮｏ）は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【０６９３】デバイスは、複数のサービス許可チケット（ＳＰＴ）に基づく複数の異なるパーティション内のファイルアクセスを許容する処理も可能である。例えば、デバイス認証の成立を条件として、複数のサービス許可チケット（ＳＰＴ）に基づく複数の異なるパーティション内のファイルアクセスを許容することが可能である。各パーティション毎のファイルアクセスルールは、アクセス制御データとして構成されるサービス許可チケット（ＳＰＴ）に記述され、デバイスは、アクセス機器から複数のサービス許可チケット（ＳＰＴ）を受領し、各チケットがデバイス認証を要求している場合は、記述に従ってデバイス認証が成立したことを条件として各パーティション内のファイルアクセスを許容する。

【０６９４】また、デバイスは、複数のサービス許可チケット（ＳＰＴ）の各々が異なる認証条件を定めている場合は、各サービス許可チケット（ＳＰＴ）に設定されたパーティション認証の認証成立を条件として、複数のサービス許可チケット（ＳＰＴ）の指定ファイルに対するアクセスを許容する。

【０６９５】次にステップＳ８５３において、ファイルアクセス装置は、デバイスに対してサービス許可チケット（ＳＰＴ：Service Parmission Ticket）を送信する。サービス許可チケット（ＳＰＴ）は、パーティションマネージャの管理下のサービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）により発行されるチケットである。サービス許可チケット（ＳＰＴ）は、デバイスに対するアクセス制御チケットであり、先に説明した図２８、図３１のデータフォーマット構成を持つチケットである。

【０６９６】なお、サービス許可チケット（ＳＰＴ）を、チケットユーザに対して送信する際には、公開鍵方式の場合、サービス許可チケット（ＳＰＴ）発行手段（SPT Issuer）の公開鍵証明書（CERT_SPTI）も一緒に送信する。ＳＰＴ発行手段の公開鍵証明書（CERT_SPTI）の属性（Attribute）は、デバイス内のＦＤＢ（File Definition Block）に記録されたチケット発行手段コード(SPTIC)と一致する。

【０６９７】ファイル登録チケット（ＳＰＴ）を受信（Ｓ８６２）したデバイスは、受信したチケット（ＳＲＴ）の正当性と利用者チェック処理を実行（Ｓ８６３）する。チケットの正当性の検証処理は、共通鍵方式によるＭＡＣ検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器（チケット利用者）の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子（図２８、図３１参照）との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット（ＰＲＴ）の適用処理についての説明中、図５７～図５９を用いて説明したと同様の処理であるので説明を省略する。

【０６９８】デバイスにおいて、受信チケット（ＳＰＴ）の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合（Ｓ８６４でＮｏ）は、サービス許可チケット（ＳＰＴ）受理エラーをファイルアクセス装置に通知（Ｓ８６８）する。チケットおよび利用者の正当なことが確認できた場合（Ｓ８６４でＹｅｓ）は、受信したサービス許可チケット（ＳＰＴ）に記述されたルールに従いデバイス内のメモリ部に格納されたファイルオープン処理が実行される。この処理の詳細については、別フローを用いて後段で詳述する。

【０６９９】サービス許可チケット（ＳＰＴ）の記述に従って、ファイルのオープン処理に成功（Ｓ８６６でＹｅｓ）すると、ＳＰＴ受理成功をファイルアクセス装置に通知（Ｓ８６７）する。一方、ファイルのオープン処理に失敗（Ｓ８６６でＮｏ）した場合は、ＳＰＴ受理エラーをファイルアクセス装置に通知（Ｓ８６８）する。

【０７００】ファイルアクセス装置は、ＳＰＴ受理結果を受信（Ｓ８５４）し、ＳＰＴ処理結果を判定し、ＳＰ

T受理結果がエラーである場合（Ｓ８５５でＮｏ）は、エラーとして処理を終了し、ＳＰＴ受理結果が成功（Ｓ８５５でＹｅｓ）である場合は、すべてのＳＰＴ送信が終了したか否かを判定（Ｓ８５６）し、未送信ＳＰＴがある場合は、ステップＳ８５３以下を繰り返し実行する。

【０７０１】すべてのＳＰＴ送信が終了した場合は、ステップＳ８５７、Ｓ８６９においてサービス許可チケット（ＳＰＴ）に従ったファイルアクセスを実行し、ファイルアクセス処理の終了の後、セッションクリアコマンドの送受信（Ｓ８５８，Ｓ８７０）を実行し、デバイス側に生成した認証テーブルを破棄（Ｓ８７１）し、処理を終了する。ファイルアクセス処理の詳細については、別フローを用いて後段で詳述する。なお、認証テーブルは、ステップＳ８５１，Ｓ８６０の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット（ＰＲＴ）の適用処理の項目において説明した構成、すなわち、図５１の構成と同様のものである。

【０７０２】このようにサービス許可チケット（ＳＰＴ）を利用して、デバイス内に設定されたパーティション内のファイルに対するアクセス処理が実行される。以下、当処理に含まれるファイルオープン処理（Ｓ８６５）、各種ファイルアクセス処理（Ｓ８５７，Ｓ８６９）について説明する。

【０７０３】（ファイルオープン処理）図８０のフローに従って、ファイルオープン処理（図７９、Ｓ８６５）について説明する。ファイルオープン処理は、デバイスが受信したサービス許可チケット（ＳＰＴ）に従って実行する処理である。

【０７０４】ステップＳ８８１において、デバイスは、受信したサービス許可チケット（ＳＰＴ）に指定されたファイルがデバイス内に生成されて存在しているか否かを判定する。サービス許可チケット（ＳＰＴ）には処理対象ファイルのファイルＩＤが記録（図２８、図３１参照）されており、同一のＩＤを持つファイルの有無を例えばファイル定義ブロック（図２４）を参照して判定する。チケットに記述されたＩＤと同一ＩＤのファイルが存在しない場合は、処理が不可能であるのでエラー終了する。

【０７０５】チケットに記述されたＩＤと同一ＩＤのファイルが存在する場合は、ステップＳ８８２において、ファイルオープンテーブルにサービス許可チケット（ＳＰＴ）に記述されたチケット発行手段（Ticket issuer＝ＰＭＣ:Partition managerCode)と、サービス許可チケット（ＳＰＴ）に記述されたファイルＩＤとを対応付けたエントリを書き込む。

【０７０６】さらに、ステップＳ８８３において、ファイルオープンテーブルに生成したエントリに対応付けてサービス許可チケット（ＳＰＴ）に記述されたファイルアクセスモード［File Access Mode :アクセスを許諾するファイル（File）へのアクセスモード（Access Mode）］を書き込み、ステップＳ８８４において、サービス許可チケット（ＳＰＴ）に記述されたアクセス許可グループファイル［Groupof Target File :アクセスを許すファイル（File）のグループ（Group）］を書き込み、ステップＳ８８５において、サービス許可チケット（ＳＰＴ）に記述されたアクセス許可ファイル識別子［Target File ID :アクセスを許すファイル（File）の識別子（ＩＤ）］の書き込み、さらにステップＳ８８６において、サービス許可チケット（ＳＰＴ）に記述されたターゲットファイル（Target File））に対する処理態様データ［Read/Write Permission : アクセスを許すファイル（File）（ターゲットファイル（Target File））に対する処理態様（読み出し（Read），書き込み（Write）の許可）］の書き込み処理を行なう。なお、ターゲットファイルに対する処理としては、読み出し（Read），書き込み（Write）に限らず、様々な処理を設定可能である。

【０７０７】ファイルオープンテーブルの構成例を図８１、図８２に示す。ファイルオープンテーブルは、デバイスにおいてアクセス処理状態にあるファイルおよびアクセスモード他の情報を記録したテーブルであり、デバイスが受信したサービス許可チケット（ＳＰＴ）の記述情報を記録してデバイスの記憶手段に格納する。

【０７０８】チケットが唯一のファイルに対してのみアクセスを許可する形式のサービス許可チケット（図２８参照）である場合は、ファイルオープンテーブルは、
*Ticket Issuer :チケット発行手段（Ticket Issuer）の識別子
* File ID :パーティション内のアクセスファイル（File）の識別子（ＩＤ）
* File Access Mode :アクセスを許諾するファイル（File）へのアクセスモード（Access Mode）
の情報を格納する。この場合のファイルオープンテーブルの構成例を図８１に示す。

【０７０９】図８１に示すように、ファイルオープンテーブルにはグループ情報であるTicket Issuer :チケット発行手段（Ticket Issuer）の識別子として、パーティションマネージャコード（ＰＭＣ）が記述され、パーティションが判別され、ファイルＩＤによりファイルが識別され、ファイルアクセスモードにより実行可能なアクセス態様（ｅｘ．読み取り（ＲＥＡＤ）、書き込み（Ｗｒｉｔｅ）、暗号化復号化（Ｅｎｃ，Ｄｅｃ））が判定可能となる。

【０７１０】また、サービス許可チケット（ＳＰＴ）がパーティションに設定されたファイル中の複数ファイルに対してアクセスを許可する形式のサービス許可チケット（図３１参照）の場合は、上記情報に加えて
* Group of Target File :アクセスを許すファイル（Fi

le）のグループ（Group）
* Target File ID :アクセスを許すファイル（File）の識別子（ＩＤ）
* Read/Write Permission : アクセスを許すファイル（File）（ターゲットファイル（Target File））に対する処理態様（読み出し（Read），書き込み（Write））の許可
の各情報がテーブルに書き込まれる。この場合のファイルオープンテーブルの構成例を図８２に示す。

【０７１１】図８２に示すように、複数ファイルに対してアクセスを許可する形式のサービス許可チケットに対応して設定されるファイルオープンテーブルには、図８１に示すデータの他に、アクセスを許すターゲットファイル（File）のグループとしてのパーティション識別データとしてのパーティションマネージャコード（ＰＭＣ）と、アクセスを許すターゲットファイル（File）の識別子（ＩＤ）としてのファイルＩＤと、ターゲットファイル（Target File））に対する処理態様を示す［Read/Write Permission］データが格納され、複数ファイルに対する実行可能な処理が判定可能となる。

【０７１２】複数のファイルに対してアクセスを実行する処理とは、例えばファイルＡに格納された鍵を用いて、ファイルＢに格納されたデータを暗号化する処理を実行する場合などである。このためには、ファイルＢはファイルＡの読み出し要求に対して許可を与える必要がある。この場合、ファイルＢのことをソースファイル、許可を与える相手のファイルのことをターゲットファイルと呼ぶ。

【０７１３】このように、デバイスは、アクセス機器とのセッション中に受領したサービス許可チケット（ＳＰＴ）に基づいて、チケット発行手段（Ticket Issuer(PMC)）としてのパーティションマネージャコード（ＰＭＣ）、ファイルオープン処理を実行したファイルの識別データとしてのファイル識別子と、サービス許可チケット（ＳＰＴ）に記述されたアクセスモードを対応付けたファイルオープンテーブルを生成し、該ファイルオープンテーブルを参照して前記アクセス機器からの受領コマンドの実行の可否の判定が可能となる。

【０７１４】（ファイルアクセス処理）次に、図７９のステップＳ８５７、Ｓ８６９において実行されるファイルアクセス処理の詳細について説明する。

【０７１５】まず、図８１に示すファイルオープンテーブルが生成された場合のアクセス処理について、図８３を用いて説明する。図の左側に２つのファイルアクセス装置（Ｒ／Ｗ：デバイスアクセス機器としてのリーダライタ）７５０、７６０を示し、右側にファイルの生成されたデバイス１００のパーティション部分を示す。

【０７１６】ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７５０は、デバイスとの相互認証の後、
ファイルＩＤ：［０ｘ０００２］
ファイルアクセスモード：［読み取り：Ｒｅａｄ］
のアクセス許可チケットをデバイス１００に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【０７１７】このとき、デバイスには図８１に示すファイルオープンテーブルの第２行のエントリが生成される。このエントリは、パーティションマネージャコード（ＰＭＣ１）で識別されるパーティション内のファイルＩＤ［０ｘ０００２］に対してアクセスモード［読み取り：Ｒｅａｄ］の処理が実行可能であることを示している。

【０７１８】このとき、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７５０は、コマンドを生成してデバイスに対して送信する。例えばファイルＩＤ［０ｘ０００２］のデータ読み取りコマンド：Ｒｅａｄ Ｃｏｍｍａｎｄ（０ｘ０００２）をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ［０ｘ０００２］に対してアクセスモード［読み取り：Ｒｅａｄ］の処理が実行可能であることを確認し、読み取り処理を実行する。

【０７１９】また、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７５０が、例えばファイルＩＤ［０ｘ０００２］のデータ書き込みコマンド：Ｗｒｉｔｅ Ｃｏｍｍａｎｄ（０ｘ０００２）、あるいはファイルＩＤ［０ｘ０００１］のデータの暗号化処理コマンド：Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１）をデバイスに送信した場合は、コマンドを受信したデバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ［０ｘ０００２］に対する［書き込み：Ｗｒｉｔｅ］の処理、および、ファイルＩＤ［０ｘ０００１］の［暗号化処理］が、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７５０から受領したサービス許可チケット（ＳＰＴ）によって許可されていないことを確認し、処理を停止する。

【０７２０】また、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７６０は、デバイスとの相互認証の後、
ファイルＩＤ：［０ｘ０００１］
ファイルアクセスモード：［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］
のアクセス許可チケットをデバイス１００に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【０７２１】このとき、デバイスには図８１に示すファイルオープンテーブルの第１行のエントリが生成される。このエントリは、パーティションマネージャコード（ＰＭＣ１）で識別されるパーティション内のファイルＩＤ［０ｘ０００１］に対してアクセスモード［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］の処理が実行可能であることを示している。

【０７２２】このとき、ファイルアクセス装置（Ｒ／

W：リーダライタ）７６０は、コマンドを生成してデバイスに対して送信する。例えばファイルＩＤ［０ｘ０００１］の暗号化コマンド［Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１）］をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ：０ｘ０００１に対してアクセスモード［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］の処理が実行可能であることを確認し、暗号化処理を実行する。

【０７２３】また、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７６０が、例えばファイルＩＤ［０ｘ０００２］のデータ読み取りコマンド：Ｒｅａｄ Ｃｏｍｍａｎｄ（０ｘ０００２）をデバイスに送信した場合は、コマンドを受信したデバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ［０ｘ０００２］に対する［読み取り：Ｒｅａｄ］の処理が、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７６０から受領したサービス許可チケット（ＳＰＴ）によって許可されていないことを確認し、処理を停止する。

【０７２４】このように、サービス許可チケットユーザであるデバイスアクセス機器としてのリーダライタからデバイスが受信したサービス許可チケット（ＳＰＴ）に基づいて、デバイスは、前述の図８０の処理フローに従ってファイルオープンテーブルを生成し、生成したファイルオープンテーブルに基づいてファイルアクセス装置であるリーダライタからの各コマンドの実行可否を決定し、決定に従って処理を実行する。

【０７２５】次に、２つのファイルに対する処理を実行する場合のアクセス処理について、図８４を用いて説明する。図の左側に２つのファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７７０、７８０を示し、右側にファイルの生成されたデバイス１００のパーティション部分を示す。

【０７２６】まず、ターゲットファイルを指定したサービス許可チケット（ＳＰＴ）（図３１参照）を使用した処理の実行例を説明する。

【０７２７】ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７７０は、デバイスとの相互認証の後、

ＳＰＴフォーマット１

ファイルＩＤ：［０ｘ０００１］

ファイルアクセスモード：［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］

および、ＳＰＴフォーマット２

ファイルＩＤ：［０ｘ０００２］

ターゲットファイル・グループ：［ＰＭＣ１］

ターゲットファイルＩＤ：［０ｘ０００１］

読書き許可：［読み取り：Ｒｅａｄ］

の２つのアクセス許可チケットをデバイス１００に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【０７２８】このとき、デバイスには図８２に示すファイルオープンテーブルのエントリが生成される。このエントリは、パーティションマネージャコード（ＰＭＣ１）で識別されるパーティション内のファイルＩＤ［０ｘ０００１］は、鍵のファイルであり、暗号化、復号化が可能になるようにオープンされる。ファイルＩＤ［０ｘ０００２］は、データのファイルであり、外部からの読み出しはファイルアクセスモード（File Access Mode）の欄が空白のため不可能であり、ファイルＩＤ［０ｘ０００１］に対して［読み取り：Ｒｅａｄ］を実行可能とする目的のためにオープンされて、ファイルオープンテーブルのエントリとして設定されていることを示している。

【０７２９】このとき、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７７０は、コマンドを生成してデバイスに対して送信する。例えばファイルＩＤ［０ｘ０００２］の読み取り、ファイルＩＤ［０ｘ０００１］による内部暗号化コマンド：Ｉｎｔｅｒｎａｌ Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１，０ｘ０００２）を送信し、デバイスがコマンドを受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ［０ｘ０００２］に対して、ターゲットファイルグループ［ＰＭＣ１］、ターゲットファイル［０ｘ０００１］の［暗号化処理］に対して［読み取り：Ｒｅａｄ］を実行可能であることを判定し、ファイルＩＤ［０ｘ００００２］のデータを読み取り、ファイルＩＤ［０ｘ００００１］の鍵（ｋｅｙ）による暗号化を実行して暗号化データをアクセス装置に送信する。

【０７３０】このターゲットファイルを指定したサービス許可チケット（ＳＰＴ）（図３１参照）を使用した処理によればあるファイルから読み出したデータを他のファイルに格納された暗号化鍵を用いて暗号化したデータを取得する処理が可能となり、復号データが外部に漏れるおそれがない。

【０７３１】次に、ターゲットファイルを指定したサービス許可チケット（ＳＰＴ）（図３１参照）ではなく、唯一のファイルに対する処理を指定したサービス許可チケット（ＳＰＴ）（図２８参照）を複数用いた場合の処理について説明する。

【０７３２】ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７８０は、デバイスとの相互認証の後、ＳＰＴフォーマット１として、

ファイルＩＤ：［０ｘ０００２］

ファイルアクセスモード：［読み取り：Ｒｅａｄ］

またＳＰＴフォーマット２として、

ファイルＩＤ：［０ｘ０００１］

ファイルアクセスモード：［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］

の２つのアクセスチケットをデバイス１００に送信し、チケットの正当性検証、チケット発行者、利用者検証に成功したとする。

【０７３３】このとき、デバイスには図８１に示すファイルオープンテーブルの第１行および第２行の各エントリが生成される。このエントリは、パーティションマネージャコード（ＰＭＣ１）で識別されるパーティション内のファイルＩＤ［０ｘ０００１］に対してアクセスモード［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］の処理が実行可能であり、パーティションマネージャコード（ＰＭＣ１）で識別されるパーティション内のファイルＩＤ：０ｘ０００２に対してアクセスモード［読み取り：Ｒｅａｄ］の処理が実行可能であることを示している。

【０７３４】このとき、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７８０は、コマンドを生成してデバイスに対して送信する。まず、ファイルＩＤ：［０ｘ０００２］のデータ読み取りコマンド：Ｒｅａｄ Ｃｏｍｍａｎｄ（０ｘ０００２）をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ：０ｘ０００２に対してアクセスモード［読み取り：Ｒｅａｄ］の処理が実行可能であることを確認し、読み取り処理を実行し、ファイルアクセス装置に読み取りデータを送信する。

【０７３５】次に、ファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７８０は、さらにコマンドを生成してデバイスに対して送信する。ファイルＩＤ［０ｘ０００１］によるデータ（Ｄａｔａ）の暗号化コマンド［Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１，Ｄａｔａ）］をデバイスが受信すると、デバイスはファイルオープンテーブルのエントリを確認し、ファイルＩＤ［０ｘ０００１］に対してアクセスモード［暗号化復号化処理：Ｅｎｃ＆Ｄｅｃ］の処理が実行可能であることを確認し、暗号化処理を実行し暗号化データ［Ｅｎｃｒｙｐｔｉｏｎ Ｄａｔａ］をファイルアクセス装置（Ｒ／Ｗ：リーダライタ）７８０に送信する。

【０７３６】このように、ターゲットファイルを指定したサービス許可チケット（ＳＰＴ）（図３１参照）ではなく、唯一のファイルに対する処理を指定したサービス許可チケット（ＳＰＴ）（図２８参照）を複数用いた場合は、暗号化対象データの読み出し処理が実行され、ファイルアクセス装置７８０とデバイス間におけるデータ転送処理の回数が増加することになる。また、データが暗号化されずにデバイスの外に読み出されてしまう。

【０７３７】一方、サービス許可チケット（ＳＰＴ）（図３１参照）に、アクセス対象とした複数のデータファイルを識別する複数のファイル識別子を含ませ、該複数のファイル識別子中、一方はターゲットファイル識別子として設定するとともにターゲットファイルに対する読み取りまたは書き込み許可データを格納し、他方のデータファイルのアクセスモードとして該データファイルに格納した暗号鍵を用いた暗号化処理を設定した構成とすれば、メモリ搭載デバイスは、アクセス機器からサービス許可チケット（ＳＰＴ）を受領して、指定アクセスモードに従った処理として、ターゲットファイルの読み取りおよび暗号鍵による暗号化処理を実行することになり、メモリ搭載デバイス内における内部暗号化処理を実行することが可能となる。また、データが暗号化されずにデバイスの外に流出することを防止することができる。

【０７３８】サービス許可チケット（ＳＰＴ）を発行するチケット発行手段は、メモリ搭載デバイスのメモリ領域を管理するエンティティであるパーティションマネージャの管理下にあるチケット発行手段であり、各アクセス機器に応じて各種のアクセスモードを設定したサービス許可チケット（ＳＰＴ）を個別に発行することにより、各アクセス機器に応じて異なる態様のアクセスを実行可能とした構成が実現される。

【０７３９】（セッション鍵の使用態様）なお、ファイルアクセス装置とデバイス間において送受信されるデータは、デバイスのユーザ情報、あるいは金額情報など外部に対する漏洩を防止すべきデータであることが多い。従ってファイルアクセス装置とデバイス間において送受信されるデータは、暗号化処理を実行し、また改竄チェック値としてのＭＡＣ（Message Authentication Code）を付加したデータとするのが望ましい。

【０７４０】データの暗号化には、ファイルアクセス装置とデバイス間において実行される相互認証処理において生成されるセッション鍵を用いることが可能である。前述したように相互認証には、デバイスに対するデバイス認証、各パーティションに対する認証としてのパーティション認証がある。パーティションに生成済みのファイルに対するアクセスを実行する場合、データ転送の際に適用する暗号化鍵としていずれを適用するかはいくつかの選択肢がある。

【０７４１】例えば図８５に示すように、デバイス１００とアクセス装置８００との間において、デバイス認証によって生成されたセッション鍵Ｋｓｅｓ１、パーティションマネージャコード（ＰＭＣ１）に対応するパーティションとのパーティション認証によって生成されたセッション鍵Ｋｓｅｓ２、パーティションマネージャコード（ＰＭＣ２）に対応するパーティションとのパーティション認証によって生成されたセッション鍵Ｋｓｅｓ３がある場合がある。

【０７４２】これらは、相互認証の際に生成される認証テーブル（図５１、図５２参照）にセッションクリアまで格納される。

【０７４３】デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（ＰＣ他の通信装置）は、これら複数のセッション鍵のいずれを適用して暗号化通信を実行するかを、ルールとして予め取り決めておき、決定されたルールに従ってセッション鍵を適用する構成が可能である。

【０７４４】複数の異なるパーティション内のファイル

アクセスを、複数の異なるパーティション各々に対応して設定された認証条件であるパーティション認証またはデバイス認証のすべての認証成立を条件として許容する場合、複数の認証処理の結果、取得された複数のセッション鍵に基づいて唯一の統合セッション鍵を生成し、該統合セッション鍵に基づいてアクセス機器との通信データの暗号処理を実行する。

【０７４５】統合セッション鍵生成手法としての１つの方法は、デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（ＰＣ他の通信装置）間における相互認証処理によって複数のセッション鍵Ｋｓｅｓ１～ＫｓｅｓＮが生成された場合、これら複数のセッション鍵Ｋｓｅｓ１～ＫｓｅｓＮの排他論理和処理（ｅｘ．８バイト処理）を実行し演算結果を通信データの暗号化用セッション鍵とする方法である。すなわち、

Kses＝Kses1 XOR Kses2 XOR Kses3…

ＸＯＲ：排他論理和処理（ｅｘ．８バイト処理）

によって算出されたＫｓｅｓをセッション鍵として用いる。

【０７４６】デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（ＰＣ他の通信装置）間では、双方の認証テーブルに格納されたセッション鍵を排他論理和演算しその出力値をセッション鍵として使用するというルールを定め、該ルールに基づいてセッション鍵を算出して通信データの暗号化に用いる構成とする。また、同様にして、相互認証時に同時に共有した別のセッション鍵、例えば公開鍵認証時において生成したセッション鍵、あるいはセッション鍵生成データ、例えばＹ座標の下位６４ビットを用いてセッション中の通信データにＭＡＣ値を付加することができる。このＭＡＣ値を通信データ（暗号化データである場合もある）とともに送信し、受信側でＭＡＣ検証処理を行なうことで、通信路上のデータ改竄を防止することが可能となる。ＭＡＣ生成、検証処理については先に説明した図５９を参照されたい。

【０７４７】あるいは、デバイスとデバイスとの通信を実行するデバイスアクセス機器としてのリーダライタ（ＰＣ他の通信装置）間における相互認証処理によって取得された複数のセッション鍵Ｋｓｅｓ１～ＫｓｅｓＮの中で、いずれかの１つの鍵（ｅｘ．最新のセッション鍵）を選択してその後の通信処理におけるデータ暗号化鍵として適用するというルールを設定し、該ルールに従って、セッション鍵を選択して通信データの暗号化に用いる構成としてもよい。

【０７４８】なお、上述した複数のセッション鍵の演算による算出、あるいは選択処理は、ファイルアクセス装置とデバイス間の暗号化通信のみならず、すべてのチケット（ＰＲＴ，ＦＲＴ，ＳＰＴ，ＤＵＴ）ユーザ（デバイスアクセス機器としてのリーダライタなどのデバイスとのデータ通信を実行する機器）とデバイス間の暗号化通信処理において、相互認証により複数のセッション鍵が生成された場合に適用できる。各チケットユーザとデバイス相互において、どのようなルールに従って、適用するセッション鍵を複数のセッション鍵から算出するかまたは選択するかについては予めルールとして取り決め、相互にルールを確認した後実行するか、あるいは各チケットにルールを記録しておくなどの措置が採用可能である。

【０７４９】次に、図８６、図８７を用いてファイルアクセス装置によるデバイスに対するアクセス処理（図７９の処理フローにおけるステップＳ８５７，８６９）手順の代表的例について説明する。

【０７５０】図８６を用いて１つのファイルのみに対してアクセスを実行する場合の処理（Ｎｏｒｍａｌ）について説明し、図８７を用いて複数ファイルに対してアクセスを行なう場合の処理（Ｃｏｍｂｉｎａｔｉｏｎ）について説明する。

【０７５１】まず、図８６の１つのファイルのみに対してアクセスを実行する場合の処理（Ｎｏｒｍａｌ）について説明する。図８６のフローにおいて、左側がファイルアクセス装置、右側がデバイス（図５参照）の処理を示す。なお、ファイルアクセス処理において、ファイルアクセス装置とデバイス間におけるデータ転送の際には、相互認証処理において取得したセッション鍵Ｋｓｅｓ、あるいは複数のセッション鍵から演算マタは選択されたセッション鍵を用いて暗号化され、また改竄チェック用のＭＡＣの生成、検証処理が実行される。

【０７５２】ファイルアクセス装置は、ステップＳ８９１において、アクセスコマンドをデバイスに送信する。このコマンドは、アクセス対象のファイルＩＤ、アクセスモードを指定したコマンドであり、例えば先に図８３を用いて説明したようなファイルＩＤ［０ｘ０００２］のデータ読み取りコマンド：Ｒｅａｄ Ｃｏｍｍａｎｄ（０ｘ０００２）、あるいは、ファイルＩＤ［０ｘ０００１］の暗号化コマンド［Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１）］などである。

【０７５３】デバイスは、ファイルアクセス装置からのコマンドを受信（Ｓ９０１）すると、コマンドに含まれるファイルＩＤ、アクセスモードがファイルオープンテーブルに許可されたエントリとして記録されているか否かを判定（Ｓ９０２）する。ファイルオープンテーブルにコマンドに対応するファイルＩＤ、アクセスモードのエントリが存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９０８）する。

【０７５４】ファイルオープンテーブルにコマンドに対応するファイルＩＤ、アクセスモードのエントリが存在した場合は、ステップＳ９０３において、デバイスのメモリ内の対応パーティションのファイル定義ブロック

（ＦＤＢ）（図２４参照）に記録されたファイルアクセス認証方式（Acceptable Authentication Type：特定のファイル（File）アクセスを行う際に、指定する認証方式）を参照し、アクセス対象のファイルに対するアクセスコマンドの実行に必要な認証レベル（公開鍵認証を必要とするか）を確認する。

【０７５５】ステップＳ９０３において、ファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定である場合は、ステップＳ９０４において、アクセスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９０８）する。認証の終了または未了は、相互認証時に設定される認証テーブル（図５１参照）に基づいて判定される。

【０７５６】ステップＳ９０３において、ファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定であり、ステップＳ９０４において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要としない設定である場合は、次に、ステップＳ９０５において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（ＦＤＢ）（図２４参照）に記録されたファイルアクセス検証方式（Acceptable Verification Type：特定のファイル（File）アクセスを行う際に、指定する検証方式）を参照し、アクセス対象のファイルに対するアクセスコマンドの実行に必要な検証レベル（公開鍵方式の検証を必要とするか）を確認する。

【０７５７】ステップＳ９０５において、ファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定である場合は、ステップＳ９０６において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９０８）する。

【０７５８】ステップＳ９０５において、ファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定であり、ステップＳ９０６において、公開鍵方式のチケット検証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要としない設定である場合は、ステップＳ９０７において、ファイルアクセス装置から受信したアクセスコマンドの処理を実行し、結果をファイルアクセス装置に送信する。

【０７５９】アクセスコマンド結果を受信（Ｓ８９２）したファイルアクセス装置は、さらに他のファイルアクセスを実行するか否かを判定（Ｓ８９３）し、他のファイルアクセスを実行する場合は、ステップＳ８９１以下を繰り返し実行し、他にファイルアクセスを実行しない場合は処理を終了する。

【０７６０】次に、図８７を用いて複数ファイルに対してアクセスを行なう場合の処理（Ｃｏｍｂｉｎａｔｉｏｎ）について説明する。図８７のフローにおいて、左側がファイルアクセス装置、右側がデバイス（図５参照）の処理を示す。なお、ファイルアクセス処理において、ファイルアクセス装置とデバイス間におけるデータ転送の際には、相互認証処理において取得したセッション鍵Ｋｓｅｓ、あるいは複数のセッション鍵から演算または選択されたセッション鍵を用いて暗号化され、また改竄チェック用のＭＡＣの生成、検証処理が実行される。

【０７６１】ファイルアクセス装置は、ステップＳ９１１において、アクセスコマンドをデバイスに送信する。このコマンドは、アクセス対象のファイルＩＤ（ソース）、ターゲットファイルＩＤ、アクセスモードを指定したコマンドであり、例えば先に図８４を用いて説明したような、ソースファイルＩＤ［０ｘ０００２］に対して、ターゲットファイルＩＤ［０ｘ０００１］の鍵による内部暗号化処理の実行を指定するコマンド［Ｉｎｔｅｒｎａｌ Ｅｎｃｒｙｐｔｉｏｎ Ｃｏｍｍａｎｄ（０ｘ０００１，０ｘ０００２）］などである。

【０７６２】デバイスは、ファイルアクセス装置からのコマンドを受信（Ｓ９２１）すると、ファイルオープンテーブルのターゲットファイルＩＤのエントリにアクセスコマンドの許可があるか否かを判定（Ｓ９２２）する。ファイルオープンテーブルのターゲットファイルＩＤのエントリにアクセスコマンドの許可が存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。

【０７６３】ファイルオープンテーブルのターゲットファイルＩＤのエントリにアクセスコマンドの許可が存在した場合は、ステップＳ９２３において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（ＦＤＢ）（図２４参照）に記録されたファイルアクセス認証方式（Acceptable Authentication Type：特定のファイル（File）アクセスを行う際に、指定する認証方式）を参照し、アクセス対象のターゲットファイルに対するアクセスコマンドの実行に必要な認証レベル（公開鍵認証を必要とするか）を確認する。

【０７６４】ステップＳ９２３において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定である場合は、ステップＳ９２４において、アク

セスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。認証の終了または未了は、相互認証時に設定される認証テーブル（図５１参照）に基づいて判定される。

【０７６５】ステップＳ９２３において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定であり、ステップＳ９２４において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要としない設定である場合は、次に、ステップＳ９２５において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（ＦＤＢ）（図２４参照）に記録されたファイルアクセス検証方式（Acceptable Verification Type：特定のファイル（File）アクセスを行う際に、指定する検証方式）を参照し、アクセス対象のターゲットファイルに対するアクセスコマンドの実行に必要な検証レベル（公開鍵方式の検証を必要とするか）を確認する。

【０７６６】ステップＳ９２５において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定である場合は、ステップＳ９２６において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。

【０７６７】ステップＳ９２５において、アクセス対象のターゲットファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定であり、ステップＳ９２６において、公開鍵方式のチケット検証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要としない設定である場合は、次に、ステップＳ９２７において、アクセスコマンドに含まれるターゲットファイルＩＤによって指定されるファイルのアクセス方法（Read/Write）をコマンドに基づいて確認する。

【０７６８】デバイスは、ファイルアクセス装置からのコマンドに含まれるソースファイルＩＤによって指定されるファイルがアクセスコマンドに含まれるアクセス方法（Read/Write）に対してオープンしているか否かを判定（Ｓ９２８）する。ファイルオープンテーブルにコマンド実行のためのアクセス方法（Read/Write）が存在しない場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。

【０７６９】ファイルオープンテーブルにコマンドに対応するアクセス方法（Read/Write）が存在した場合は、ステップＳ９２９において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（ＦＤＢ）（図２４参照）に記録されたファイルアクセス認証方式（Acceptable Authentication Type：特定のファイル（File）アクセスを行う際に、指定する認証方式）を参照し、アクセス対象のソースファイルに対するアクセスコマンドの実行に必要な認証レベル（公開鍵認証を必要とするか）を確認する。

【０７７０】ステップＳ９２９において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定である場合は、ステップＳ９３０において、アクセスコマンドに必要な認証レベルの認証としての公開鍵認証が済んでいるか否かを判定し、認証未了の場合は、コマンドに従った処理を実行せず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。認証の終了または未了は、相互認証時に設定される認証テーブル（図５１参照）に基づいて判定される。

【０７７１】ステップＳ９２９において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable Authentication Type）が公開鍵認証を必要とする設定であり、ステップＳ９３０において、公開鍵認証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス認証方式（Acceptable AuthenticationType）が公開鍵認証を必要としない設定である場合は、次に、ステップＳ９３１において、デバイスのメモリ内の対応パーティションのファイル定義ブロック（ＦＤＢ）（図２４参照）に記録されたファイルアクセス検証方式（Acceptable Verification Type：特定のファイル（File）アクセスを行う際に、指定する検証方式）を参照し、アクセス対象のソースファイルに対するアクセスコマンドの実行に必要な検証レベル（公開鍵方式の検証を必要とするか）を確認する。

【０７７２】ステップＳ９３１において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定である場合は、ステップＳ９３２において、アクセスコマンドに必要な検証レベルの検証としての公開鍵方式のチケット検証が済んでいるか否かを判定し、検証未了の場合は、コマンドに従った処理を実行せ

ず、アクセスエラーをファイルアクセス装置に送信（Ｓ９３４）する。

【０７７３】ステップＳ９３１において、アクセス対象のソースファイルに対して設定されたファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要とする設定であり、ステップＳ９３２において、公開鍵方式のチケット検証が済んでいると判定された場合、あるいは、ファイル定義ブロック（ＦＤＢ）のファイルアクセス検証方式（Acceptable Verification Type）が公開鍵方式のチケット検証を必要としない設定である場合は、ステップＳ９３３において、ファイルアクセス装置から受信したアクセスコマンドの処理を実行し、結果をファイルアクセス装置に送信する。

【０７７４】アクセスコマンド結果を受信（Ｓ９１２）したファイルアクセス装置は、さらに他のファイルアクセスを実行するか否かを判定（Ｓ９１３）し、他のファイルアクセスを実行する場合は、ステップＳ９１１以下を繰り返し実行し、他にファイルアクセスを実行しない場合は処理を終了する。

【０７７５】上述したファイルアクセス処理は、ファイル内にある１つのファイル構造によって指定されるデータが格納された場合の処理を想定して説明しているが、異なるファイル構造データを１つのファイル内に格納し、１つのファイルに対する１つのコマンドにより、上述の複数ファイルに対するシーケンシャル処理と同様の処理を実行する構成も可能である。

【０７７６】図８８に１つのファイルに対する１つのコマンドにより、１ファイル内のデータに対してシーケンシャル処理を実行する構成を説明する図を示す。

【０７７７】ファイルは図に示すように、電子マネーファイルであり、金額データとしての［Ｐｕｒｓｅ］、利用ログデータとしての「Ｌｏｇ」、データに対する暗号化または復号用の鍵データとしての［Ｋｅｙ］から構成される。

【０７７８】例えば、図８８（ａ）に示すように、預け入れコマンド（Deposit Command）を規定し、ファイル内の金額データとしての［Ｐｕｒｓｅ］にＸ円を加算（Ｓ９４１）し、さらに、ファイル内の利用ログデータとしての「Ｌｏｇ」に［Ｐｕｒｓｅ］にＸ円を加算した記録を書き込む（Ｓ９４２）という２つの処理を実行させる構成とすることが可能である。

【０７７９】先に説明したファイルアクセスモード（図２９参照）の入金系に対応する許容コマンド（図３０参照）として、上述の預け入れコマンド（Deposit Command）を定義し、アクセス許可チケットのファイルアクセスモード（File Access Mode）に［入金系］を設定し、ファイルＩＤ(File ID)として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット（ＳＰＴ）を生成して、ファイルアクセス装置からデバイスに対して送信した後、預け入れコマンド（DepositCommand）とともに、預け入れ金額データを送信することにより、図８８（ａ）に示すようなデバイスにおいて１つのファイル内のデータに対するシーケンシャル処理を実行させることが可能となる。

【０７８０】また、図８８（ｂ）に示すように、レシート生成コマンド（Make Receipt Command）を規定し、ファイル内の金額データとしての［Ｐｕｒｓｅ］からＸ円を減算（Ｓ９４５）し、さらに、ファイル内の利用ログデータとしての「Ｌｏｇ」に［Ｐｕｒｓｅ］からＸ円を減算した記録を書き込み（Ｓ９４６）、さらに「Ｌｏｇ」にデータに対する暗号化鍵データとしての［Ｋｅｙ］を適用して署名をつけて送信する（Ｓ９４７）という３ステップの処理を実行させる構成とすることも可能である。

【０７８１】この場合はファイルアクセスモード（図２９参照）の出金系に対応する許容コマンド（図３０参照）として、上述のレシート生成コマンド（Make Receipt Command）を定義し、アクセス許可チケットのファイルアクセスモード（File Access Mode）に［出金系］を設定し、ファイルＩＤ(File ID)として、電子マネーを構成する複合ファイルを指定したアクセス許可チケット（ＳＰＴ）を生成して、ファイルアクセス装置からデバイスに対して送信した後、レシート生成コマンド（Make Receipt Command）とともに、引き出し金額データを送信することにより、図８８（ｂ）に示すようなデバイスにおいて１つのファイル内のデータに対するシーケンシャル処理を実行させることが可能となる。

【０７８２】このようにデバイスは、サービス許可チケット（ＳＰＴ）に指定された処理ファイルが複合ファイルである場合、アクセス機器からの受領コマンドの処理対象ファイルを複合ファイル内から選択して処理を実行する。アクセス機器からのデータ処理コマンドが一連の複数の処理を含むシーケンス処理コマンドである場合は、デバイスは、シーケンス処理コマンドに含まれる各コマンドの処理対象ファイルを、サービス許可チケット（ＳＰＴ）によって指定された複合ファイル内から順次選択して実行する。

【０７８３】［Ｂ４．７．サービス許可チケット（ＳＰＴ）を利用したアクセス処理各方式における処理手順］

上述したサービス許可チケット（ＳＰＴ）を利用したアクセス処理ファイルの設定登録処理において、パーティションマネージャが管理し、サービス許可チケット（ＳＰＴ）ユーザであるデバイスアクセス機器としてのリーダライタとデバイス間において、相互認証が実行され、サービス許可チケット（ＳＰＴ）に基づくファイルアクセスがなされる。相互認証処理の態様は、公開鍵相互認証、共通鍵相互認証の２種類のいずれかであり、またチケット（ＳＰＴ）の検証処理も公開鍵系の署名検証、共通鍵系のＭＡＣ検証の２種類のいずれかが実行されるこ

とになる。すなわち処理態様としては大きく分けて、
（A）相互認証（公開鍵）、チケット（ＳＰＴ）検証（公開鍵）
（B）相互認証（公開鍵）、チケット（ＳＰＴ）検証（共通鍵）
（C）相互認証（共通鍵）、チケット（ＳＰＴ）検証（共通鍵）
（D）相互認証（共通鍵）、チケット（ＳＰＴ）検証（公開鍵）
の４態様がある。

【０７８４】これらの４態様についての処理を、認証局（ＣＡ（ＰＭ））、パーティションマネージャ（ＰＭ）、ＳＰＴチケットユーザであるデバイスアクセス機器としてのリーダライタ、デバイス、各エンティティ間において実行されるデータ転送処理を中心として図を用いて簡潔に説明する。

【０７８５】（A）相互認証（公開鍵）、チケット（ＳＰＴ）検証（公開鍵）

まず、相互認証処理に公開鍵方式を適用し、チケット（ＳＰＴ）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図８９を用いて説明する。

【０７８６】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

（１）パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）の発行、パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）は、パーティションマネージャ（ＰＭ）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））から発行される。なお、本構成は、パーティションマネージャがサービス許可チケット発行手段（ＳＰＴ Issuer）を兼ねる構成であり、サービス許可チケット発行手段（ＳＰＴ Issuer）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用する構成である。

【０７８７】（２）サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）の発行、サービス許可チケットユーザ（ＳＰＴ User：具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ））の公開鍵証明書（Ｃｅｒｔ．Ｒ／Ｗ）は、サービス許可チケットユーザ（ＳＰＴ User）であるリーダライタ（Ｒ／Ｗ）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））によって発行される。なお、パーティションマネージャがサービス許可チケットユーザ（ＳＰＴ User）を兼ねる構成も可能であり、その場合は、サービス許可チケットユーザ（ＳＰＴ User）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０７８８】（３）サービス許可チケット（ＳＰＴ）の生成処理

サービス許可チケット（ＳＰＴ）は、パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の秘密鍵による署名(Signature)が生成（図１２参照）されてＳＰＴに付加される。

【０７８９】（４）ＳＰＴおよびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）のサービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴTicket Issuer）により発行されたサービス許可チケット（ＳＰＴ）は、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）とともにサービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）に対して送信される。

【０７９０】（５）デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証

サービス許可チケットユーザ（ＳＰＴ User）であるリーダライタは、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の発行したサービス許可チケット（ＳＰＴ）に従ったファイルアクセスを実行しようとする対象のデバイスに対し、チケットユーザ（ＳＰＴ User）としてのリーダライタ（Ｒ／Ｗ）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）をデバイスに送信し、公開鍵方式の相互認証（図５０参照）を実行する。

【０７９１】（６）ＳＰＴおよびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）のデバイスに対する供給

デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証が成立すると、チケットユーザ（ＳＰＴ User）としてのリーダライタ（Ｒ／Ｗ）は、デバイスに対してサービス許可チケット（ＳＰＴ）、およびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）を送信する。

【０７９２】デバイスは、受信したサービス許可チケット（ＳＰＴ）について、（１）チケット発行者（Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）が改竄されたものでない正当な

公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）の公開鍵証明書（ＣＥＲＴＰＭ）のオプション領域に記録されたコードと、デバイス内のＦＤＢ（File Definition Block）に記録された（SPTIC）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＳＰＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＳＰＴチケットに格納されたＳＰＴユーザ（チケットユーザとしてのリーダライタ）とチケットユーザ（ＳＰＴ User）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）名（ＤＮ）の一致を確認し、相互認証済みであることを確認することによりＳＰＴユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０７９３】（７）ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット（ＳＰＴ）に記述されたルールに従ってアクセスを実行する。

【０７９４】以上の処理によって、相互認証（公開鍵）、チケット（ＳＰＴ）検証（公開鍵）の各方式に従ったファイルアクセス処理が実行される。

【０７９５】（Ｂ）相互認証（公開鍵）、チケット（ＳＰＴ）検証（共通鍵）

次に、相互認証処理に公開鍵方式を適用し、チケット（ＳＰＴ）検証に共通鍵方式を適用する場合の各エンティテイ間のデータ転送について図９０を用いて説明する。

【０７９６】図に示す番号順に各エンティテイ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【０７９７】（１）サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）の発行、サービス許可チケットユーザ（ＳＰＴ User：具体的には、デバイスに対してチケットを送信するデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ））の公開鍵証明書（Ｃｅｒｔ．Ｒ／Ｗ）は、サービス許可チケットユーザ（ＳＰＴ User）であるリーダライタ（Ｒ／Ｗ）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））によって発行される。なお、パーティションマネージャがサービス許可チケットユーザ（ＳＰＴ User）を兼ねる構成も可能であり、その場合は、サービス許可チケットユーザ（ＳＰＴ User）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０７９８】（２）サービス許可チケット（ＳＰＴ）の生成処理

サービス許可チケット（ＳＰＴ）は、パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）により生成される。この場合、共通鍵方式の検証値としてＭＡＣ(Message Authentication Code)（図５９参照）がＳＰＴに付加される。

【０７９９】（３）ＳＰＴのサービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴTicket Issuer）により発行されたサービス許可チケット（ＳＰＴ）は、サービス許可チケットユーザ（ＳＰＴ User）としてのリーダライタ（Ｒ／Ｗ）に対して送信される。

【０８００】（４）リーダライタ（Ｒ／Ｗ）とデバイス間の相互認証

サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタは、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の発行したサービス許可チケット（ＳＰＴ）に従ったファイルアクセスを実行しようとする対象のデバイスに対し、チケットユーザ（ＳＰＴ User）としてのリーダライタ（Ｒ／Ｗ）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）をデバイスに送信し、公開鍵方式の相互認証（図５０参照）を実行する。パーティションマネージャ（ＰＭ）の公開鍵証明書を使用可能である。

【０８０１】（５）ＳＰＴのデバイスに対する供給

デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証が成立すると、サービス許可チケットユーザ（ＳＰＴ User）であるリーダライタは、デバイスに対してサービス許可チケット（ＳＰＴ）を送信する。デバイスは、受信したサービス許可チケット（ＳＰＴ）についてＭＡＣ検証処理を実行し、ＳＰＴ発行者（SPT Issuer）の検証、さらに、ＳＰＴチケットに格納されたＳＰＴユーザ（チケットユーザとしてのリーダライタ）とチケットユーザ（ＳＰＴ User）の公開鍵証明書（Ｃｅｒｔ．ＲＷ）の識別データ（ＤＮ）として記録された識別子またはカテゴリまたはシリアル（ＳＮ）名（ＤＮ）の一致を確認し、相互認証済みであることを確認することによりＳＰＴユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０８０２】（６）ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット（ＳＰＴ）に記述されたルールに従ってアクセスを実行する。

【０８０３】以上の処理によって、相互認証（公開鍵）、チケット（ＳＰＴ）検証（共通鍵）の各方式に従ったファイルアクセス処理が実行される。

【０８０４】（Ｃ）相互認証（共通鍵）、チケット（ＳＰＴ）検証（共通鍵）

次に、相互認証処理に共通鍵方式を適用し、チケット（ＳＰＴ）検証に共通鍵方式を適用する場合の各エンティティ間のデータ転送について図９１を用いて説明する。

【０８０５】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

【０８０６】（１）サービス許可チケット（ＳＰＴ）の生成処理

サービス許可チケット（ＳＰＴ）は、パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）により生成される。この場合、共通鍵方式の検証値としてＭＡＣ(Message Authentication Code)（図５９参照）がＳＰＴに付加される。

【０８０７】（２）ＳＰＴのサービス許可チケットユーザ（ＳＰＴ User）に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴTicket Issuer）により発行されたサービス許可チケット（ＳＰＴ）は、サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタに対して送信される。

【０８０８】（３）デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証

サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）は、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の発行したサービス許可チケット（ＳＰＴ）に従ったファイルを生成しようとする対象のデバイスとの間で、共通鍵方式の相互認証（図５３、図５４参照）を実行する。

【０８０９】（４）ＳＰＴのデバイスに対する供給

デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証が成立すると、サービス許可チケットユーザ（ＳＰＴ User）であるリーダライタは、デバイスに対してサービス許可チケット（ＳＰＴ）を送信する。デバイスは、受信したサービス許可チケット（ＳＰＴ）についてＭＡＣ検証処理を実行し、ＳＰＴ発行者（SPT Issuer）の検証、さらに、ＳＰＴチケットに格納されたＳＰＴユーザ（チケットユーザとしてのリーダライタ）とチケットユーザ（ＳＰＴ User）の識別子の一致を確認し、相互認証済みであることを確認することによりＳＰＴユーザ（デバイスアクセス機器としてのリーダライタ）の検証（図５７、図５８参照）を実行する。

【０８１０】（５）ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット（ＳＰＴ）に記述されたルールに従ってアクセスを実行する。

【０８１１】以上の処理によって、相互認証（共通鍵）、チケット（ＳＰＴ）検証（共通鍵）の各方式に従ったファイルアクセス処理が実行される。

【０８１２】（Ｄ）相互認証（共通鍵）、チケット（ＳＰＴ）検証（公開鍵）

次に、相互認証処理に共通鍵方式を適用し、チケット（ＳＰＴ）検証に公開鍵方式を適用する場合の各エンティティ間のデータ転送について図９２を用いて説明する。

【０８１３】図に示す番号順に各エンティティ間でデータ転送が実行される。以下、各番号に従って処理を説明する。

（１）パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）の発行、パーティションマネージャ（ＰＭ）の公開鍵証明書（Ｃｅｒｔ．ＰＭ）は、パーティションマネージャ（ＰＭ）からの発行要求により、登録局（ＲＡ）を介した証明書発行手続きによってパーティション対応認証局（ＣＡ（ＰＡＲ））から発行される。なお、本構成は、パーティションマネージャがサービス許可チケット発行手段（ＳＰＴ Issuer）を兼ねる構成であり、サービス許可チケット発行手段（ＳＰＴ Issuer）の公開鍵証明書としてパーティションマネージャ（ＰＭ）の公開鍵証明書を使用する構成である。

【０８１４】（２）サービス許可チケット（ＳＰＴ）の生成処理

サービス許可チケット（ＳＰＴ）は、パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）により生成される。この場合、公開鍵方式の署名生成、検証を実行するため、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の秘密鍵による署名(Signature)が生成（図１２参照）されてＳＰＴに付加される。

【０８１５】（３）ＳＰＴおよびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）のサービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）に対する供給

パーティションマネージャの管理するサービス許可チケット発行手段（ＳＰＴTicket Issuer）により発行されたサービス許可チケット（ＳＰＴ）は、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）とともにサービス許可チケットユーザ（ＳＰＴ User）すなわち、デバイスに対してチケットを送信する機器（ｅｘ．デバイスアクセス機器としてのリーダライタ）に対して送信される。

【０８１６】（４）デバイスアクセス機器としてのリーダライタ（Ｒ／Ｗ）とデバイス間の相互認証

サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタは、サービス許可チケット発行手段（ＳＰＴ Ticket Issuer）の発行

したサービス許可チケット（ＳＰＴ）に従ったファイルアクセスを実行しようとする対象のデバイスとの間で、共通鍵方式の相互認証（図５３、図５４参照）を実行する。

【０８１７】（５）ＳＰＴおよびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）のデバイスに対する供給

リーダライタ（Ｒ／Ｗ）とデバイス間の相互認証が成立すると、サービス許可チケットユーザ（ＳＰＴ User）であるデバイスアクセス機器としてのリーダライタは、デバイスに対してサービス許可チケット（ＳＰＴ）、およびサービス許可チケット発行手段（ＳＰＴ Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）を送信する。

【０８１８】デバイスは、受信したサービス許可チケット（ＳＰＴ）について、（１）チケット発行者（Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）が改竄されたものでない正当な公開鍵証明書（ＣＥＲＴ）であること、（２）チケット発行者（Ticket Issuer）としてのパーティションマネージャ公開鍵証明書（Ｃｅｒｔ．ＰＭ）のオプション領域に記録されたコードと、デバイス内のＦＤＢ（File Definition Block）に記録されたチケット発行手段コード（SPTIC）の一致、（３）チケット発行手段（Ticket Issuer）がリボークされていないこと、（４）受信チケット（ＳＰＴ）の署名（Signature）の検証によりチケットが改竄のないことの確認を実行し、さらに、ＳＰＴチケットに格納されたＳＰＴユーザ（チケットユーザとしてのリーダライタ）とチケットユーザ（ＳＰＴ User）の識別子の一致を確認し、相互認証済みであることを確認することによりＳＰＴユーザ（リーダライタ）の検証（図５７、図５８参照）を実行する。

【０８１９】（６）ファイルアクセス

デバイスは、処理対象ファイルにサービス許可チケット（ＳＰＴ）に記述されたルールに従ってアクセスを実行する。

【０８２０】以上の処理によって、相互認証（共通鍵）、チケット（ＳＰＴ）検証（公開鍵）の各方式に従ったファイルアクセス処理が実行される。

【０８２１】［Ｂ５．データアップデートチケット（ＤＵＴ）を利用したデバイスのデータ更新処理］次に、データアップデートチケット（ＤＵＴ：Data Update Ticket）を利用したデバイスのデータ更新処理について説明する。データアップデートチケット（ＤＵＴ：Data Update Ticket）は、デバイスに格納された様々なデータの更新処理を実行する際に適用されるアクセスコントロールチケットである。正当なデータアップデートチケット（ＤＵＴ）発行手段（Ticket Issuer）の発行したＤＵＴを用い、ＤＵＴに記録された手続きに従ってチケットユーザ（ｅｘ．デバイスアクセス機器としてのリーダライタ）によりデバイスにアクセスすることで、ＤＵＴに記録された制限内でデータ処理を実行することができる。

【０８２２】なお、前述したように、データアップデートチケット（ＤＵＴ：Data UpdateTicket）は、デバイスマネージャの管理するデータ項目の更新処理を実行するために適用されるチケットＤＵＴ（ＤＥＶ）と、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットＤＵＴ（ＰＡＲ）（図３２参照）がある。

【０８２３】デバイスに格納したデータにデータアップデートチケット（ＤＵＴ）を適用してデータ更新を実行する処理について説明する。図９３以下のフロー他の図面を参照して説明する。なお、データ更新処理には、デバイスとデータ更新を実行するデバイスアクセス機器としてのリーダライタ間における相互認証処理（デバイス認証またはパーティション認証）、データアップデートチケット（ＤＵＴ：Data Update Ticket）の正当性検証処理が含まれる。

【０８２４】図９３に示すデータ更新処理フローについて説明する。図９３において、左側がデータ更新装置、右側がデバイス（図５参照）の処理を示す。なお、データ更新装置は、デバイスに対するデータ読み取り書き込み処理可能な装置（ｅｘ．デバイスアクセス機器としてのリーダライタ、ＰＣ）であり、図１０のデバイスアクセス機器としてのリーダライタに相当する。まず、図９３を用いて、データ更新処理の概要を説明し、その後、当処理に含まれるデータ更新操作の詳細を図９４のフローを用いて説明する。

【０８２５】まず、図９３のステップＳ９５１とＳ９６０において、データ更新装置とデバイス間にでの相互認証処理が実行される。データ送受信を実行する２つの手段間では、相互に相手が正しいデータ通信者であるか否かを確認して、その後に必要なデータ転送を行なうことが行われる。相手が正しいデータ通信者であるか否かの確認処理が相互認証処理である。相互認証処理時にセッション鍵の生成を実行して、生成したセッション鍵を共有鍵として暗号化処理を実行してデータ送信を行なう。

【０８２６】相互認証処理については、先のパーティション生成、削除処理の欄で説明したと同様の処理であり、デバイス認証またはパーティション認証のいずれかが実行される。それぞれについて共通鍵方式認証、あるいは公開鍵方式認証処理のいずれかが適用される。この相互認証処理は、前述の図４８～図５６を用いて説明したと同様の処理であるので説明を省略する。

【０８２７】なお、相互認証処理として実行すべき処理は、適用するデータアップデートチケット（ＤＵＴ）（図３２参照）の* Authentication Type :デバイス（Device）の相互認証のタイプ（公開鍵認証、または、共

通鍵認証、または、いずれでも可（Any））によって決定される。

【０８２８】認証処理に失敗した場合（Ｓ９５２，Ｓ９６１でＮｏ）は、相互が正当な機器、デバイスであることの確認がとれないことを示し、以下の処理は実行されずエラーとして処理は終了する。

【０８２９】認証処理に成功すると、データ更新装置は、デバイスに対してデータアップデートチケット（ＤＵＴ：Data Update Ticket）を送信する。データアップデートチケット（ＤＵＴ）は、デバイスマネージャまたはパーティションマネージャの管理下のデータアップデートチケット（ＤＵＴ）発行手段（DUT Issuer）により発行されるチケットである。データアップデートチケット（ＤＵＴ）は、デバイスに対するアクセス制御チケットであり、先に説明した図３２のデータフォーマット構成を持つチケットである。

【０８３０】なお、データアップデートチケット（ＤＵＴ）を、チケットユーザに対して送信する際には、公開鍵方式の場合、データアップデートチケット（ＤＵＴ）発行手段（DUT Issuer）の公開鍵証明書（CERT_DUTI）も一緒に送信する。ＤＵＴ発行手段の公開鍵証明書（CERT_DUTI）の属性（Attribute）は、デバイス内のＤＫＤＢ（ＰＵＢ）(Device Key Definition Block)に記録されたチケット発行手段コード（DUTIC_DEV）やＰＫＤＢ（ＰＵＢ）(Partition Key Definition Block)に記録されたチケット発行手段コード（DUTIC_PAR）の識別子(DUTIC)と一致する。

【０８３１】データアップデートチケット（ＤＵＴ）を受信（Ｓ９６２）したデバイスは、受信したチケット（ＤＵＴ）の正当性と利用者チェック処理を実行（Ｓ９６３）する。チケットの正当性の検証処理は、共通鍵方式によるＭＡＣ検証、あるいは公開鍵方式による署名検証処理のいずれかを適用して実行される。利用者チェックは、チケットを送信してきた機器（チケット利用者）の正当性をチェックする処理であり、相互認証が成立済みであり、認証相手の識別データと、チケットに記録されているチケットユーザ識別子（図３２参照）との一致等を検証する処理として実行される。これらの処理は、先のパーティション登録チケット（ＰＲＴ）の適用処理についての説明中、図５７～図５９を用いて説明したと同様の処理であるので説明を省略する。

【０８３２】デバイスにおいて、受信チケット（ＤＵＴ）の正当性と利用者チェック処理の結果、チケットおよび利用者の正当なことが確認できなかった場合（Ｓ９６４でＮｏ）は、データアップデートチケット（ＤＵＴ）受理エラーをデータ更新装置に通知（Ｓ９６８）する。チケットおよび利用者の正当なことが確認できた場合（Ｓ９６４でＹｅｓ）は、受信したデータアップデートチケット（ＤＵＴ）に記述されたルールに従いデバイス内のメモリ部に格納されたデータ（図３３参照）の更新処理を実行する。この処理の詳細については、別フローを用いて後段で詳述する。

【０８３３】データアップデートチケット（ＤＵＴ）の記述に従って、データの更新処理に成功（Ｓ９６６でＹｅｓ）すると、ＤＵＴ受理成功をデータ更新装置に通知（Ｓ９６７）する。一方、データの更新処理に失敗（Ｓ９６６でＮｏ）した場合は、ＤＵＴ受理エラーをデータ更新装置に通知（Ｓ９６８）する。

【０８３４】データ更新装置は、ＤＵＴ受理結果を受信（Ｓ９５４）し、ＤＵＴ処理結果を判定し、ＤＵＴ受理結果がエラーである場合（Ｓ９５５でＮｏ）は、エラーとして処理を終了し、ＤＵＴ受理結果が成功（Ｓ９５５でＹｅｓ）である場合はセッションクリアコマンドの送受信（Ｓ９５６，Ｓ９６９）を実行し、デバイス側に生成した認証テーブルを破棄（Ｓ９７０）し、処理を終了する。認証テーブルは、ステップＳ９５１，Ｓ９６０の相互認証処理において生成されるテーブルであり、前述したパーティション登録チケット（ＰＲＴ）の適用処理の項目において説明した構成、すなわち、図５１の構成と同様のものである。

【０８３５】このようにデータアップデートチケット（ＤＵＴ）を利用して、デバイス内に格納されたデータの更新処理が実行される。以下、当処理に含まれるデータ更新操作（Ｓ９６５）について、図９４を用いて説明する。

【０８３６】図９４の処理フローは、データアップデートチケット（ＤＵＴ）を受理したデバイスにおいて実行される処理であり、データアップデートチケット（ＤＵＴ）を送信してきた機器との相互認証が成立し、チケットの検証にも成功した以後に実行される。

【０８３７】まず、ステップＳ９７１において、デバイスは、データアップデートチケット（ＤＵＴ）の更新される古いデータのコード（Old Data Code）から更新対象データのバージョンを検索する。バージョンは、例えば更新対象がデバイスマネージャコード（ＤＭＣ）であれば、デバイス管理情報ブロック（図１５参照）にバージョンが記録され、また、パーティションマネージャコード（ＰＭＣ）であれば、パーティション管理情報ブロック（図２０参照）にバージョンが記録されている。また、パーティション登録チケット（ＰＲＴ）発行手段（ＰＲＴ Issuer）のバージョンはデバイス定義ブロック（図１６参照）に含まれる。さらに、リボケーションリスト（ＩＲＬ ＤＥＶ，ＣＲＬ ＤＥＶ）などは、リボケーションリスト中にバージョン情報が含まれる。このように情報に応じてバージョン情報の格納先が決まっており、デバイスは、更新される古いデータのコード（Old Data Code）から更新対象データのバージョンを検索する。

【０８３８】次に、デバイスは、ステップＳ９７２において、データアップデートチケット（ＤＵＴ）に記録さ

れたデータ更新をする時のバージョン条件 [Data Version Rule] を参照し、設定が［Ａｎｙ］であるか否かを判定する。

【０８３９】前述したように、データ更新をする時のバージョン条件 [Data Version Rule] は、Any, Exact, Older の３種類が存在する。Any はバージョン (Version) 条件に無関係でデータ更新が可能、Exact は、続く [Data Version Condition] に指定された値と同じ場合にデータ更新が可能、Older は、New Data Versionの方が新しい場合にのみデータ更新が可能となる。なお、バージョン条件 [DataVersion Rule] がAny,または Older の場合は、 [Data Version Condition] は使用しないかもしくは無視する。

【０８４０】データアップデートチケット（ＤＵＴ）の [Data Version Rule] の設定が［Ａｎｙ］でない場合は、バージョン条件 [Data Version Rule] に従った処理を実行する。このステップがＳ９７３～Ｓ９７５である。

【０８４１】ステップＳ９７３では、データアップデートチケット（ＤＵＴ）のバージョン条件 [Data Version Rule] を参照し、設定が［ＥＸＡＣＴ］であるか否かを判定する。［ＥＸＡＣＴ］は、 [Data Version Condition] に指定された値と同じ場合にデータ更新が可能なことを示す。設定が［ＥＸＡＣＴ］である場合、ステップＳ９７４で、更新対象データ［Ｏｌｄ　Ｄａｔａ］のバージョンがデータアップデートチケット（ＤＵＴ）の [Data Version Condition] に記録されたバージョン値と一致するか否かを判定する。一致する場合にのみ次ステップに進み、一致しない場合は、更新処理を実行せずエラー終了とする。

【０８４２】ステップＳ９７３で、データアップデートチケット（ＤＵＴ）のバージョン条件 [Data Version Rule] が［ＥＸＡＣＴ］でないと判定された場合は、設定は［Ｏｌｄｅｒ］である。［Ｏｌｄｅｒ］の設定は、更新対象データ［Ｏｌｄ　Ｄａｔａ］のバージョンより、データアップデートチケット（ＤＵＴ）の新規データ [New Data] のバージョンを示す [New Data Version] の方が新しい場合にのみ更新をする設定である。この［Ｏｌｄｅｒ］の設定の場合、ステップＳ９７５において、更新対象データ［Ｏｌｄ　Ｄａｔａ］のバージョンより、データアップデートチケット（ＤＵＴ）の新規データ [New Data] のバージョンを示す [New Data Version] の方が新しいか否かを判定し、新しい場合にのみ次ステップに進み、一致しない場合は、更新処理を実行せずエラー終了とする。

【０８４３】次にデバイスは、ステップＳ９７６において、データアップデートチケット（ＤＵＴ）の [Encrypted Flag] を検証する。 [Encrypted Flag] は、更新されるデータが暗号化されているか否か(暗号化：Encrypted /非暗号化：none)を示すデータである。 [Encrypted Flag] が更新対象データが非暗号化データであることを示している場合は、ステップＳ９７７において、データアップデートチケット（ＤＵＴ）の新規データ [New Data] をデバイスのメモリ部に格納された更新対象旧データ [Old Data] に置き換える処理を実行し、処理終了とする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット（ＤＵＴ：Data Update Ticket）に格納されている更新するデータのバージョン（New Data Version）を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【０８４４】また、ステップＳ９７６において、データアップデートチケット（ＤＵＴ）の [Encrypted Flag] が、更新されるデータが暗号化されている(暗号化：Encrypted)ことを示していると判定された場合は、ステップＳ９７８において、データアップデートチケット（ＤＵＴ）の [Ticket Type] を検証する。 [Ticket Type] は、チケット（Ticket）の種別（ＤＵＴ（ＤＥＶ）／ＤＵＴ（ＰＡＲ））を示すデータである。ＤＵＴ（ＤＥＶ）は、データアップデートチケット（ＤＵＴ）が、デバイスマネージャの管理するデータ項目の更新処理を実行するチケットあることを示し、ＤＵＴ（ＰＡＲ）は、パーティションマネージャの管理するパーティション内のデータ項目の更新処理を実行するために適用されるチケットであることを示してる。

【０８４５】チケットタイプ [Ticket Type] が、ＤＵＴ（ＤＥＶ）を示している場合、ステップＳ９７９～Ｓ９８２を実行し、ＤＵＴ（ＰＡＲ）を示している場合、ステップＳ９８３～Ｓ９８６を実行する。

【０８４６】チケットタイプ [Ticket Type] が、ＤＵＴ（ＤＥＶ）を示している場合、ステップＳ９７９において、データアップデートチケット（ＤＵＴ（ＤＥＶ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_DEV2（データ更新用暗号鍵）であるか否かを判定する。

【０８４７】データアップデートチケット（ＤＵＴ（ＤＥＶ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）である場合は、ステップＳ９８０において、デバイスのデバイス鍵領域（図１８参照）に格納されたKdut_DEV4（データ更新用暗号鍵）を用いて、データアップデートチケット（ＤＵＴ（ＤＥＶ））に格納された新規データ [New Data] としてのKdut_DEV1、Kdut_DEV2を復号し、デバイスのデバイス鍵領域に格納されたKdut_DEV1、Kdut_DEV2に上書きする。なお、データアップデートチケット（ＤＵＴ（ＤＥＶ））に格納

されている更新するデータのバージョン（New Data Version）を、デバイス内の更新データに対応して設定されているバージョン格納領域、この場合は、デバイスのデバイス鍵領域（図１８参照）に格納する処理を併せて実行する。

【０８４８】次に、ステップＳ９８１において、デバイスのデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）と、Kdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）とのスワップ、すなわち入れ替え処理を行ない、また、Kdut_DEV2（データ更新用暗号鍵）と、Kdut_DEV4（データ更新用暗号鍵）とのスワップ、すなわち入れ替え処理を行なって処理を終了する。

【０８４９】なお、Kdut_DEV1と、Kdut_DEV3とのスワップ、および、Kdut_DEV2と、Kdut_DEV4とのスワップ処理によって、常にKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV4（データ更新用暗号鍵）のペアがKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）のペアよりも新しいバージョンのものに維持され、書き換え対象を、常にKdut_DEV1、Kdut_DEV2として設定した処理が可能となる。

【０８５０】なお、ステップＳ９７９において、データアップデートチケット（ＤＵＴ（ＤＥＶ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）でない場合は、ステップＳ９８２において、デバイスのデバイス鍵領域（図１８参照）に格納されたKdut_DEV2（データ更新用暗号鍵）を用いて、データアップデートチケット（ＤＵＴ（ＤＥＶ））に格納された新規データ［New Data］を復号し、データアップデートチケット（ＤＵＴ（ＤＥＶ））のOld Data code（更新される古いデータのコード）の示すエリアに上書きする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット（ＤＵＴ（ＤＥＶ））に格納されている更新するデータのバージョン（New Data Version）を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【０８５１】一方、ステップＳ９７８において、チケットタイプ［Ticket Type］が、ＤＵＴ（ＰＡＲ）を示している場合、ステップＳ９８３～Ｓ９８６を実行する。

【０８５２】チケットタイプ［Ticket Type］が、ＤＵＴ（ＰＡＲ）を示している場合、ステップＳ９８３において、データアップデートチケット（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）または、Kdut_PAR2（データ更新用暗号鍵）であるか否かを判定する。

【０８５３】データアップデートチケット（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_PAR2（データ更新用暗号鍵）である場合は、ステップＳ９８４において、デバイスのパーティション鍵領域（図２３参照）に格納されたKdut_PAR4（データ更新用暗号鍵）を用いて、データアップデートチケット（ＤＵＴ（ＰＡＲ））に格納された新規データ［New Data］としてのKdut_PAR1、Kdut_PAR2を復号し、デバイスのパーティション鍵領域に格納されたKdut_PAR1、Kdut_PAR2に上書きする。なお、データアップデートチケット（ＤＵＴ（ＰＡＲ））に格納されている更新するデータのバージョン（New Data Version）を、デバイス内の更新データに対応して設定されているバージョン格納領域、この場合は、デバイスのパーティション鍵領域（図２３参照）に格納する処理を併せて実行する。

【０８５４】次に、ステップＳ９８５において、デバイスのパーティション鍵領域（図２３参照）に格納されたKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）と、Kdut_PAR3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）とのスワップ、すなわち入れ替え処理を行ない、また、Kdut_PAR2（データ更新用暗号鍵）と、Kdut_PAR4（データ更新用暗号鍵）とのスワップ、すなわち入れ替え処理を行なって処理を終了する。

【０８５５】なお、Kdut_PAR1と、Kdut_PAR3とのスワップ、および、Kdut_ PARV2と、Kdut_PAR4とのスワップ処理によって、常にKdut_PAR3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_PAR4（データ更新用暗号鍵）のペアがKdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_PAR2（データ更新用暗号鍵）のペアよりも新しいバージョンのものに維持され、書き換え対象を、常にKdut_PAR1、Kdut_PAR2として設定した処理が可能となる。

【０８５６】なお、ステップＳ９８３において、データアップデートチケット（ＤＵＴ（ＰＡＲ））に記述されたOld Data Code（更新される古いデータのコード）の示すデータがデバイス鍵領域（図１８参照）に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）でない場合は、ステップＳ９８６において、デバイスのパーティション鍵領域（図２３参照）に格納されたKdut_PAR2（データ更新用暗号鍵）を用いて、データアップデートチケット（ＤＵＴ（ＰＡＲ））に格納された

新規データ［NewData］を復号し、データアップデートチケット（ＤＵＴ（ＰＡＲ））のOld DataCode（更新される古いデータのコード）の示すエリアに上書きする。なお、更新対象データに対してバージョンが付加されている場合は、データアップデートチケット（ＤＵＴ（ＰＡＲ））に格納されている更新するデータのバージョン（New Data Version）を、デバイス内の更新データに対応して設定されているバージョン格納領域に格納する処理を実行する。

【０８５７】以上の処理がデバイスにおいて実行されるデータフップデートチケットに基づくデータ更新操作である。

【０８５８】上述したフローから理解されるように、更新対象データがデバイス鍵領域に格納された
Kdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）
Kdut_DEV2（データ更新用暗号鍵）
または、パーティション鍵領域に格納された
Kdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）
Kdut_PAR2（データ更新用暗号鍵）
である場合には、他の更新処理と異なる処理を実行する。

【０８５９】これらのKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）、Kdut_PAR1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_PAR2（データ更新用暗号鍵）についての更新処理を簡潔にまとめた図を図９５に示し、処理について説明する。図９５の（１）～（３）の順に説明する。なお、処理は、Kdut_DEV1,2と、Kdut_PAR1,2とで同様のものであるので、Kdut_DEV1,2を更新する場合について説明する。

【０８６０】（１）データアップデートチケット（ＤＵＴ）に格納する新規データ［New Data］としてのKdut_DEV1、Kdut_DEV2をデバイスのデバイス鍵領域（図１８参照）に格納されたKdut_DEV4（データ更新用暗号鍵）を用いて暗号化した後、データアップデートチケット（ＤＵＴ）に格納し、データアップデートチケット（ＤＵＴ）をデバイスに送信する。このとき、Kdut_DEV1、Kdut_DEV2を更新できるチケット発行者はKdut_DEV3、Kdut_DEV4を知らなくてはならない。

【０８６１】（２）データアップデートチケット（ＤＵＴ）を受信したデバイスは、デバイスのデバイス鍵領域に格納されたKdut_DEV4（データ更新用暗号鍵）を用いて、データアップデートチケット（ＤＵＴ）の格納新規データ［New Data］としてのKdut_DEV1、Kdut_DEV2を復号し、デバイスのデバイス鍵領域に格納されたKdut_DEV1、Kdut_DEV2に上書きする。

【０８６２】（３）次に、デバイスは、デバイスのデバイス鍵領域（図１８参照）に新規に格納されたKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）と、以前に格納済みのKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）とのスワップ、すなわち入れ替え処理を行なう。さらに、新規に格納されたKdut_DEV2（データ更新用暗号鍵）と、以前に格納済みのKdut_DEV4（データ更新用暗号鍵）とのスワップ、すなわち入れ替え処理を行なう。

【０８６３】この、Kdut_DEV1と、Kdut_DEV3とのスワップ、および、Kdut_DEV2と、Kdut_DEV4とのスワップ処理によって、常にKdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV4（データ更新用暗号鍵）のペアがKdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）のペアよりも新しいバージョンのものに維持される。つまり、Kdut_DEV1と、Kdut_DEV2の鍵は常に使用される鍵で、Kdut_DEV3と、Kdut_DEV4は、非常時にKdut_DEV1と、Kdut_DEV2を更新するとともに、現在使用されているKdut_DEV1と、Kdut_DEV2の鍵に置き換えられるバックアップ用の鍵としての役割がある。

【０８６４】なお、Kdut_DEV1（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV2（データ更新用暗号鍵）はペアとして使用され、また、Kdut_DEV3（データアップデートチケット（ＤＵＴ）のＭＡＣ検証用鍵）、Kdut_DEV4（データ更新用暗号鍵）もペアとして使用される。

【０８６５】以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０８６６】なお、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。

【０８６７】例えば、プログラムは記録媒体としてのハードディスクやＲＯＭ（Read OnlyMemory)に予め記録しておくことができる。あるいは、プログラムはフロッピーディスク、ＣＤ－ＲＯＭ(Compact Disc Read Only Memory)，ＭＯ(Magneto optical)ディスク，ＤＶＤ(Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバ

ブル記録媒体は、いわゆるパッケージソフトウエアとして提供することができる。

【０８６８】なお、プログラムは、上述したようなリムーバブル記録媒体からコンピュータにインストールする他、ダウンロードサイトから、コンピュータに無線転送したり、ＬＡＮ(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

【０８６９】なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【０８７０】

【発明の効果】上述したように、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、複数のパーティションに分割されたメモリ領域のアクセスに対して、様々な種類のアクセス制御チケットを各デバイスまたはパーティション管理エンティテイの管理の下に発行し、各チケットに記述されたルールに基づく処理をメモリ搭載デバイスにおいて実行する構成が可能となり、各パーティション内データの独立した管理構成が実現される。

【０８７１】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、パーティション対応の認証、デバイス対象の認証を公開鍵、共通鍵のいずれか指定方式に従って実行することが可能なデバイスとし、様々な環境下においてデバイスおよびアクセス装置間のセキュアなデータ通信が実行可能となる。

【０８７２】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、メモリ搭載デバイスのメモリ部は、データファイルを格納し、パーティションマネージャによって管理されるメモリ領域としての１以上のパーティション領域と、該メモリ搭載デバイスの管理者としてのデバイスマネージャによって管理されるデバイスマネージャ管理領域とを有し、メモリ部に対するアクセス制御チケットとして、デバイスマネージャの管理するアクセス制御チケット、またはパーティションマネージャの管理するアクセス制御チケットをアクセス機器から受領し、受領チケットの記述に応じて処理を実行する構成とし、実行すべき相互認証態様、アクセス制御チケットの検証態様を指定し、各態様に従って処理を可能としたので様々な環境下においてデバイスおよびアクセス装置間のセキュアなデータ通信が実行可能となる。

【０８７３】さらに、本発明のメモリアクセス制御システム、デバイス管理装置、パーティション管理装置、メモリ搭載デバイス、およびメモリアクセス制御方法、並びにプログラム記憶媒体によれば、デバイスマネージャ、パーティションマネージャの管理の下、パーティション登録チケット（ＰＲＴ）、ファイル登録チケット、サービス許可チケット（ＳＰＴ）、データアップデートチケット（ＤＵＴ）を発行し、それぞれ認証、チケット検証の成立を条件としてデバイスでの処理を実行する構成としたので、様々な処理態様に応じたサービスの提供、データ管理が各サービス主体の管理の下に実行可能となる。

【図面の簡単な説明】

【図１】本発明のシステム構成の概要を説明するシステム構成概略図（その1）である。

【図２】本発明のシステム構成の概要を説明するシステム構成概略図（その２）である。

【図３】本発明のシステム構成の具体例を説明するシステム構成概略図（その３）である。

【図４】本発明のシステムにおけるアクセス制御チケットの発行手段および利用手段との関係を説明する図である。

【図５】本発明のシステムにおけるメモリ部を有するデバイス構成を示す図である。

【図６】本発明のデバイスのメモリフォーマットを示す図である。

【図７】本発明のシステムにおけるデバイスマネージャ構成を示す図である。

【図８】本発明のシステムにおけるデバイスマネージャの制御手段構成を示す図である。

【図９】本発明のシステムにおけるパーティションマネージャ構成を示す図である。

【図１０】本発明のシステムにおけるリーダライタ（Ｒ／Ｗ）構成を示す図である。

【図１１】本発明のシステムにおいて利用可能な公開鍵証明書のフォーマットを説明する図である。

【図１２】本発明のシステムにおいて利用可能な公開鍵方式の署名生成処理フローを示す図である。

【図１３】本発明のシステムにおいて利用可能な公開鍵方式の署名検証処理フローを示す図である。

【図１４】本発明のデバイスにおけるメモリ部に格納されるデータ中の製造情報ブロックのデータ構成を示す図である。

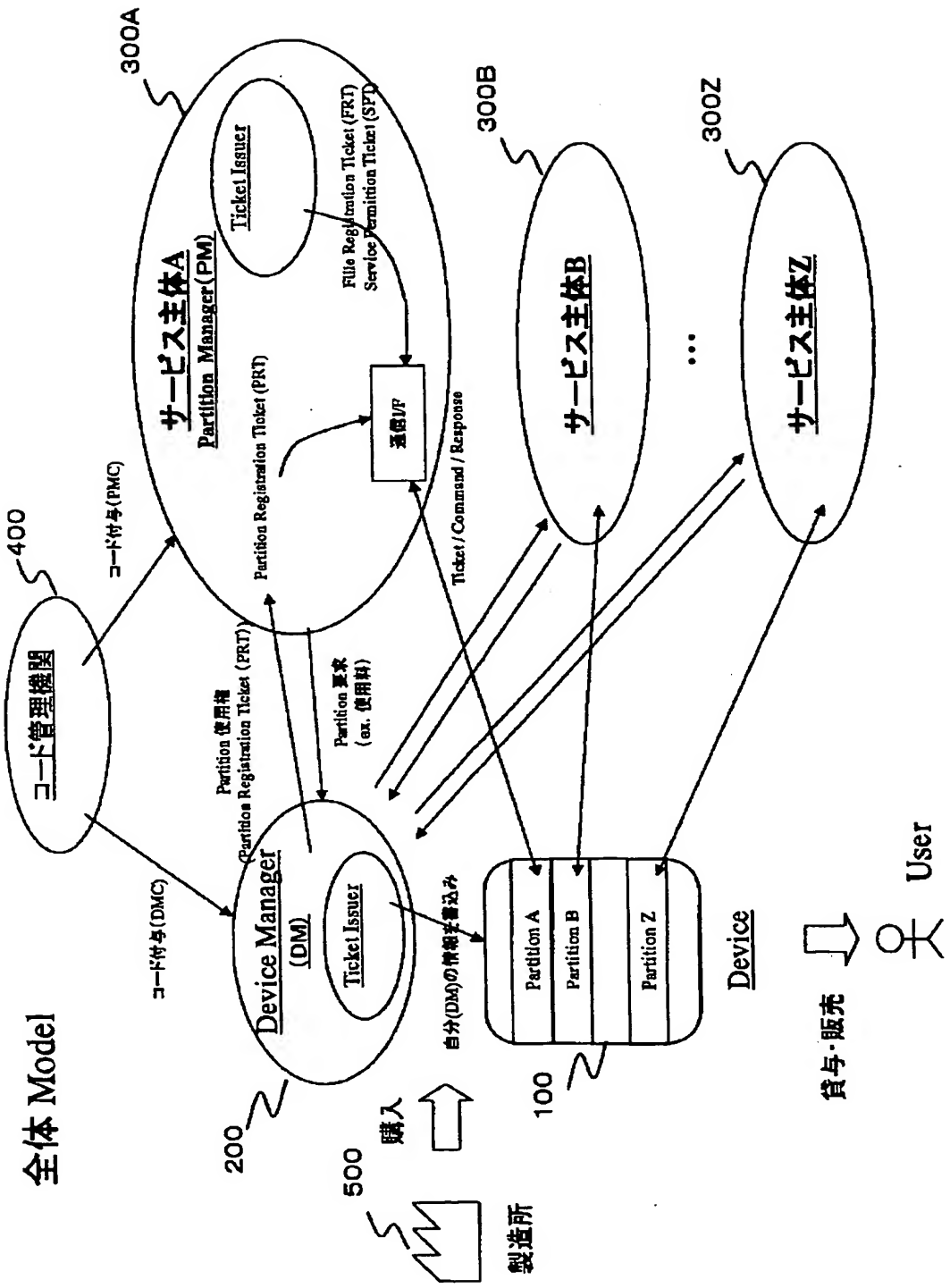
【図１５】本発明のデバイスにおけるメモリ部に格納されるデータ中のデバイス管理情報ブロックのデータ構成を示す図である。

【図１６】本発明のデバイスにおけるメモリ部に格納されるデータ中の公開鍵系デバイス鍵定義ブロックのデータ構成を示す図である。
【図１７】本発明のデバイスにおけるメモリ部に格納されるデータ中の共通鍵系デバイス鍵定義ブロックのデータ構成を示す図である。
【図１８】本発明のデバイスにおけるメモリ部に格納されるデータ中のデバイス鍵領域のデータ構成を示す図である。
【図１９】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション定義ブロックのデータ構成を示す図である。
【図２０】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション管理情報ブロックのデータ構成を示す図である。
【図２１】本発明のデバイスにおけるメモリ部に格納されるデータ中の公開鍵系パーティション鍵定義ブロックのデータ構成を示す図である。
【図２２】本発明のデバイスにおけるメモリ部に格納されるデータ中の共通鍵系パーティション鍵定義ブロックのデータ構成を示す図である。
【図２３】本発明のデバイスにおけるメモリ部に格納されるデータ中のパーティション鍵領域のデータ構成を示す図である。
【図２４】本発明のデバイスにおけるメモリ部に格納されるデータ中のファイル定義ブロックのデータ構成を示す図である。
【図２５】本発明のデバイスにおけるメモリ部に格納されるデータ中のファイルの構造のタイプについて説明する図である。
【図２６】本発明のシステムにおいて適用されるアクセス制御チケットとしてのパーティション登録チケット（ＰＲＴ）のフォーマットを示す図である。
【図２７】本発明のシステムにおいて適用されるアクセス制御チケットとしてのファイル登録チケット（ＦＲＴ）のフォーマットを示す図である。
【図２８】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット（ＳＰＴ）のフォーマット（例１）を示す図である。
【図２９】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット（ＳＰＴ）を利用したファイルアクセスのモードの種別を説明する図である。
【図３０】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット（ＳＰＴ）を利用したアクセス対象となるファイル構造を説明する図である。
【図３１】本発明のシステムにおいて適用されるアクセス制御チケットとしてのサービス許可チケット（ＳＰＴ）のフォーマット（例２）を示す図である。
【図３２】本発明のシステムにおいて適用されるアクセス制御チケットとしてのデータアップデートチケット（ＤＵＴ）のフォーマットを示す図である。
【図３３】本発明のシステムにおいて適用されるアクセス制御チケットとしてのデータアップデートチケット（ＤＵＴ）を利用した更新対象となるデータを説明する図である。
【図３４】本発明のシステムにおけるデバイス利用までの処理概略を説明する図である。
【図３５】本発明のシステムにおけるデバイス製造エンティテイによるデバイスの初期登録処理フローを示す図である。
【図３６】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー（その1）を示す図である。
【図３７】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー（その２）を示す図である。
【図３８】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー（その３）を示す図である。
【図３９】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー（その４）を示す図である。
【図４０】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理フロー（その５）を示す図である。
【図４１】本発明のシステムにおけるデバイスマネージャによるデバイスの初期登録処理の後のデバイスの格納データを説明する図である。
【図４２】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理フロー（その1）を示す図である。
【図４３】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理フロー（その２）を示す図である。
【図４４】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理を説明する図である。
【図４５】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理を説明する図である。
【図４６】本発明のシステムにおけるデバイスマネージャによる公開鍵証明書発行処理後のデバイスの格納データを説明する図である。
【図４７】本発明のシステムにおけるデバイスに対するパーティション生成、削除処理フローを示す図である。
【図４８】本発明のシステムにおけるデバイスとの相互認証処理について説明するフロー（その１）である。
【図４９】本発明のシステムにおけるデバイスとの相互認証処理（デバイス認証）について説明するフロー（その２）である。

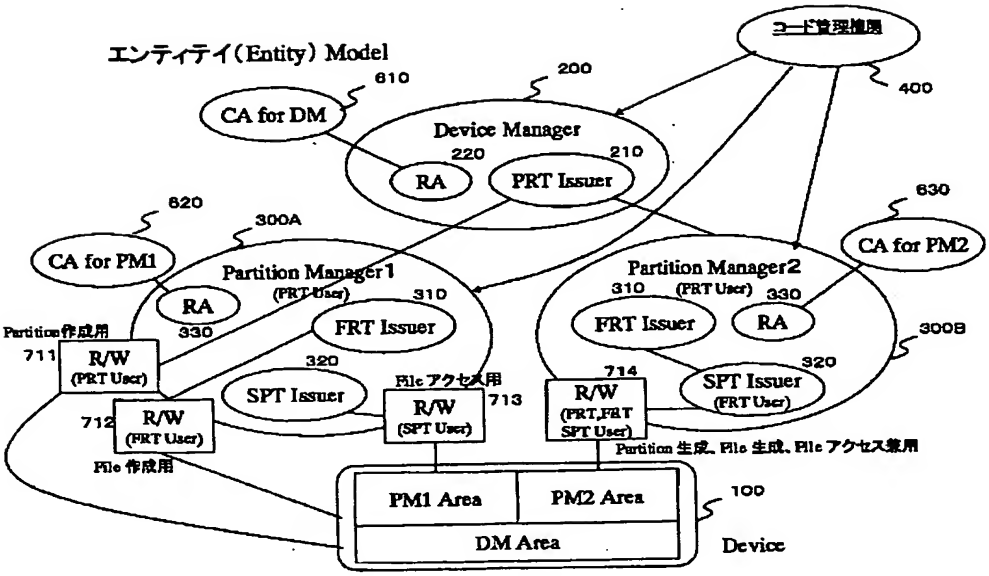
【図５０】本発明のシステムにおけるデバイスとの公開鍵方式の相互認証処理について説明する図である。
【図５１】本発明のシステムにおけるデバイスとの相互認証処理後にデバイスに生成される認証テーブルの構成を説明する図である。
【図５２】本発明のシステムにおけるデバイスとの相互認証処理後にリーダライタに生成される認証テーブルの構成を説明する図である。
【図５３】本発明のシステムにおけるデバイスとの共通鍵方式の相互認証処理について説明する図である。
【図５４】本発明のシステムにおけるデバイスとの共通鍵方式の相互認証処理について説明する図である。
【図５５】本発明のシステムにおけるデバイスとの相互認証処理（パーティション認証）について説明するフロー（その３）である。
【図５６】本発明のシステムにおけるデバイスとの相互認証処理（パーティション認証）について説明するフロー（その４）である。
【図５７】本発明のシステムにおけるチケットの正当性、利用者チェック処理について説明するフロー（その１）である。
【図５８】本発明のシステムにおけるチケットの正当性、利用者チェック処理について説明するフロー（その２）である。
【図５９】本発明のシステムにおけるチケットの正当性で適用可能なＭＡＣ生成方式について説明するフロー（その１）である。
【図６０】本発明のシステムにおけるパーティションの作成、削除操作について説明するフロー（その１）である。
【図６１】本発明のシステムにおけるパーティションの作成、削除操作について説明するフロー（その２）である。
【図６２】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その１）である。
【図６３】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その２）である。
【図６４】本発明のシステムにおけるパーティションの初期登録処理について説明するフロー（その３）である。
【図６５】本発明のシステムにおけるパーティションの初期登録処理後のデバイス格納データについて説明する図である。
【図６６】本発明のシステムにおけるパーティションマネージャによる公開鍵証明書発行処理を説明する図（その１）である。
【図６７】本発明のシステムにおけるパーティションマネージャによる公開鍵証明書発行処理を説明する図（その２）である。
【図６８】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図６９】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７０】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７１】本発明のシステムにおけるパーティションマネージャによるパーティション生成処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７２】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）を適用したファイル生成消去処理について説明するフロー図である。
【図７３】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）を適用したファイル生成削除操作について説明するフロー図である。
【図７４】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）を適用したファイル生成後のデバイス格納データを説明する図である。
【図７５】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）によるファイル生成処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７６】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）によるファイル生成処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７７】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）によるファイル生成処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７８】本発明のシステムにおけるファイル登録チケット（ＦＲＴ）によるファイル生成処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図７９】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルアクセス処理フローを示す図である。
【図８０】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルオープン操作フローを示す図である。
【図８１】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルオープン操作により

生成されるファイルオープンテーブル構成を説明する図（例１）である。
【図８２】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルオープン操作により生成されるファイルオープンテーブル構成を説明する図（例２）である。
【図８３】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルアクセス処理例を説明する図（例１）である。
【図８４】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルアクセス処理例を説明する図（例２）である。
【図８５】本発明のシステムにおける認証により生成されるセッション鍵の取扱について説明する図である。
【図８６】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルアクセス処理例を説明するフロー図（例１）である。
【図８７】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用したファイルアクセス処理例を説明するフロー図（例２）である。
【図８８】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）を適用した複合ファイルのアクセス処理例を説明する図である。
【図８９】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）によるファイルアクセス処理において、公開鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図９０】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）による処理において、公開鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図９１】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）による処理において、共通鍵方式認証、共通鍵方式チケット検証を実行した場合の処理を説明する図である。
【図９２】本発明のシステムにおけるサービス許可チケット（ＳＰＴ）による処理において、共通鍵方式認証、公開鍵方式チケット検証を実行した場合の処理を説明する図である。
【図９３】本発明のシステムにおけるデータアップデートチケット（ＤＵＴ）によるデータ更新処理フローを示す図である。
【図９４】本発明のシステムにおけるデータアップデートチケット（ＤＵＴ）によるデータ更新操作フローを示す図である。
【図９５】本発明のシステムにおけるデータアップデートチケット（ＤＵＴ）によるデータ更新処理例を説明する図である。
【図９６】従来のメモリ構造を示す図である。
【図９７】従来のメモリ管理者、利用者の関係を説明する図である。
【図９８】従来のメモリ領域確保処理について説明する図である。
【図９９】従来のメモリアクセス方式について説明する図である。
【符号の説明】
１００　デバイス
２００　デバイスマネージャ
３００　パーティションマネージャ
４００　コード管理機関
５００　デバイス製造エンティテイ
２１０　パーティション登録チケット（ＰＲＴ）発行手段
２２０　登録局（ＲＡ）
３１０　ファイル登録チケット（ＦＲＴ）発行手段
３２０　サービス許可チケット（ＳＰＴ）発行手段
３３０　登録局（ＲＡ）
６１０，６２０　認証局（ＣＡ）
７１１～７１４　リーダライタ（Ｒ／Ｗ）
７２１～７２２　チケットユーザ
１０１　ＣＰＵ
１０２　通信ＩＦ
１０３　ＲＯＭ
１０４　ＲＡＭ
１０５　暗号処理部
１０６　メモリ部（ＥＥＰＲＯＭ）
２１１，２２１　制御手段
２１２，２２２　データベース
２１１１　制御部
２１１２　ＲＯＭ
２１１３　ＲＡＭ
２１１４　表示部
２１１５　入力部
２１１６　ＨＤＤ
２１１７　ドライブ
２１１８　通信Ｉ／Ｆ
３１１，３２１，３３１　制御手段
３１２，３２２，３３２　データベース
７０１　ＣＰＵ
７０２　通信ＩＦ
７０３　ＲＯＭ
７０４　ＲＡＭ
７０５　暗号処理部
７０６　メモリ部（ＥＥＰＲＯＭ）

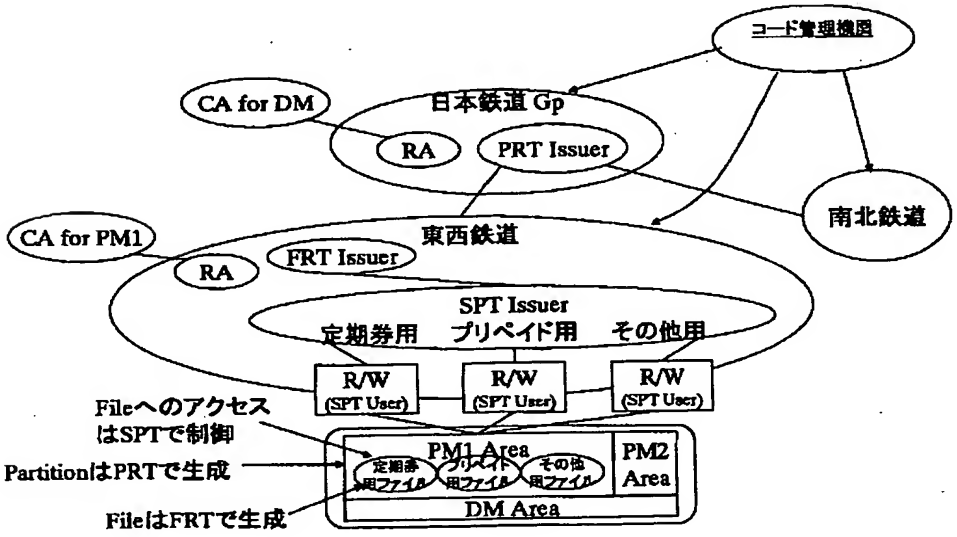
【図１】


全体 Model
300A
サービス主体A
Partition Manager(PM)
Ticket Issuer
File Registration Ticket (FRT)
Service Permition Ticket (SPT)
通信I/F
Partition Registration Ticket (PRT)
Ticket / Command / Response
300B
サービス主体B
300Z
サービス主体Z
400
コード管理機関
コード付与(PMC)
コード付与(DMC)
Partition 使用権
(Partition Registration Ticket (PRT))
Partition 要求
(ex. 使用料)
Device Manager
(DM)
Ticket Issuer
200
自分(DM)の情報を書込み
500
購入
製造所
Partition A
Partition B
Partition Z
Device
100
貸与・販売
User

【図２】



【図３】



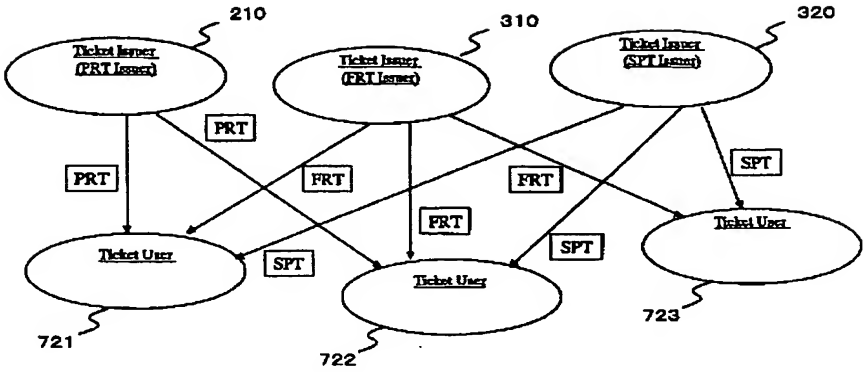
【図８１】

File Open テーブル(1)

Group	File ID	File Access Mode
PMC1	0x0001	Enc, Dec
PMC1	0x0002	Read

【図４】

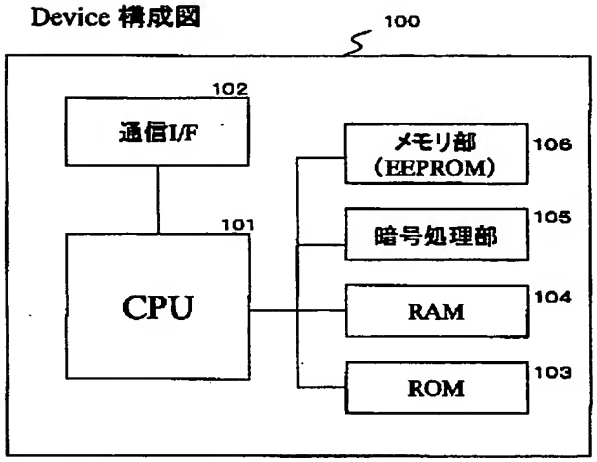
Ticket, Ticket Issuer & Ticket User



Partition Registration Ticket (PRT)
File Registration Ticket (FRT)
Service Permittion Ticket (SPT)

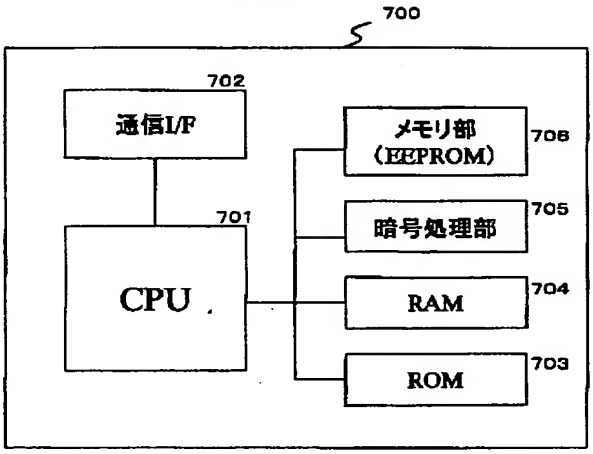
【図５】

Device 構成図



【図１０】

Reader/Writer 構成図

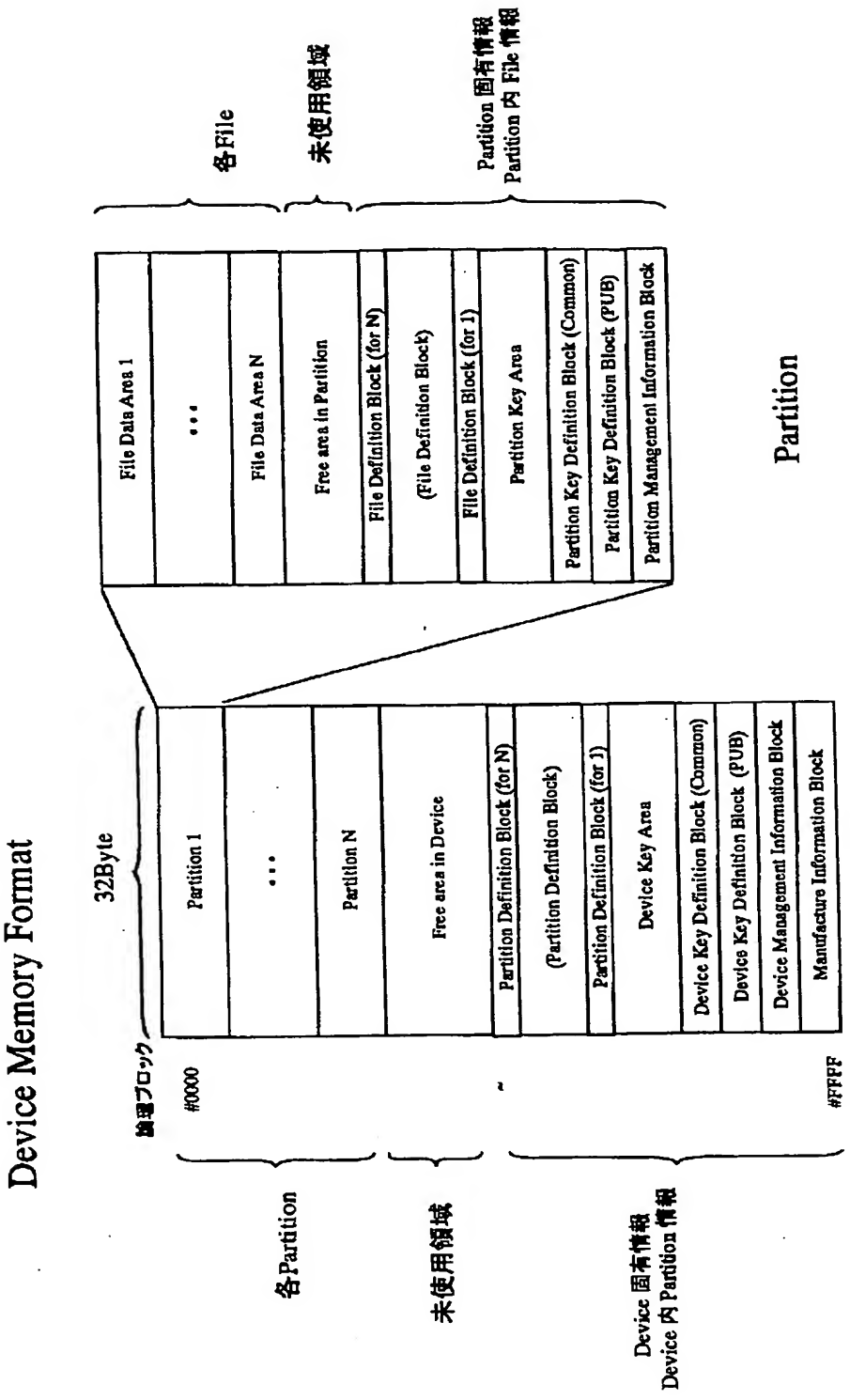


【図１１】

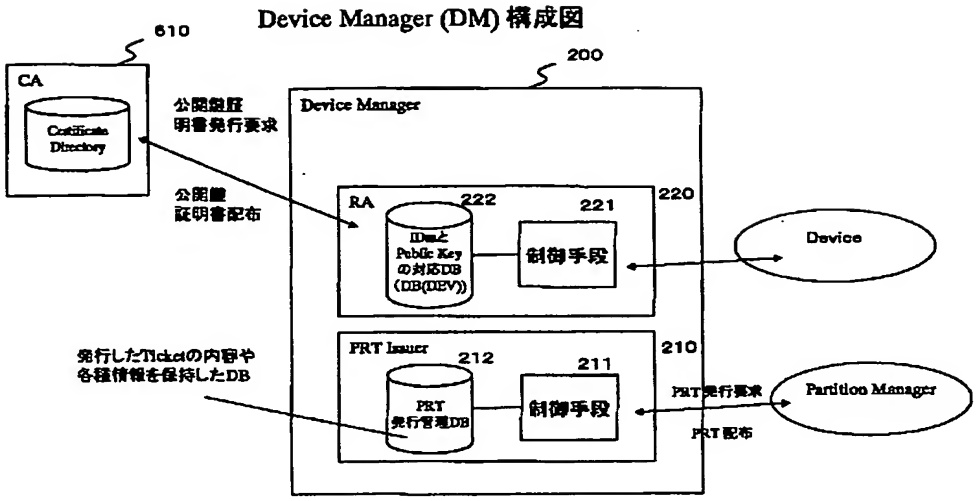
Certificate Format

証明書のバージョン番号
発行局(認証局)が割り付ける証明書通し番号(SN)
署名アルゴリズム識別子フィールド:アルゴリズムとパラメータ
発行局(認証局)の名前
証明書の有効期限フィールド:開始日時、終了日時
公開鍵証明書の利用者名(Subject)
利用者の公開鍵フィールド:鍵アルゴリズムと鍵情報(鍵そのもの)
オプション領域 (属性など)
発行局(認証局)署名

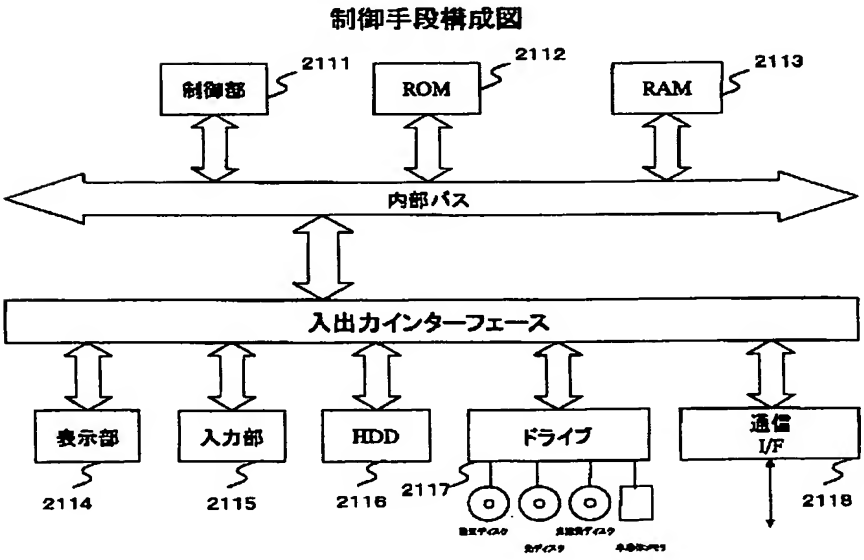
【図6】


Device Memory Format
32Byte
論理ブロック
#0000
#FFFF
Partition 1
Partition N
Free area in Device
Partition Definition Block (for N)
(Partition Definition Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (PUB)
Device Management Information Block
Manufacture Information Block
Device
各Partition
未使用領域
Device 固有情報
Device 内 Partition 情報
File Data Area 1
File Data Area N
Free area in Partition
File Definition Block (for N)
(File Definition Block)
File Definition Block (for 1)
Partition Key Area
Partition Key Definition Block (Common)
Partition Key Definition Block (PUB)
Partition Management Information Block
Partition
各File
未使用領域
Partition 固有情報
Partition 内 File 情報

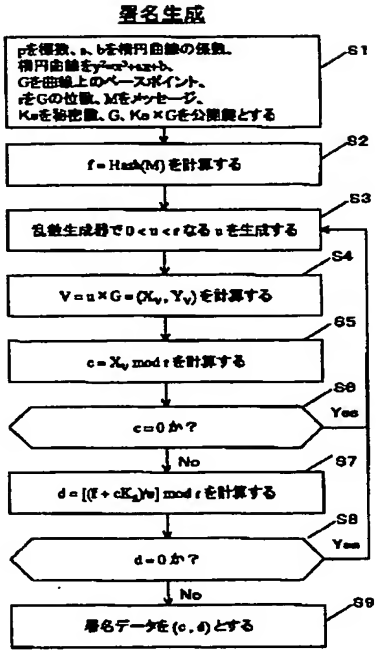
【図７】



【図８】

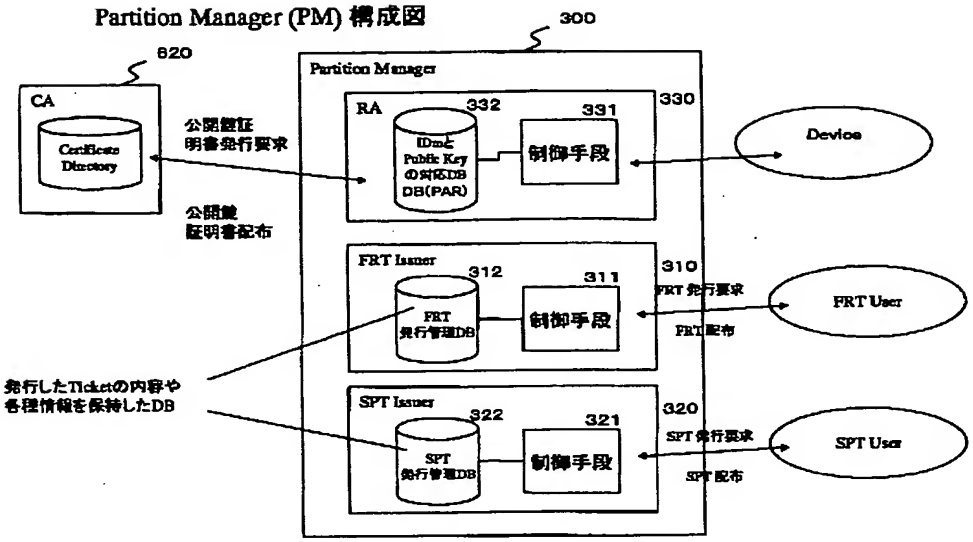


【図１２】

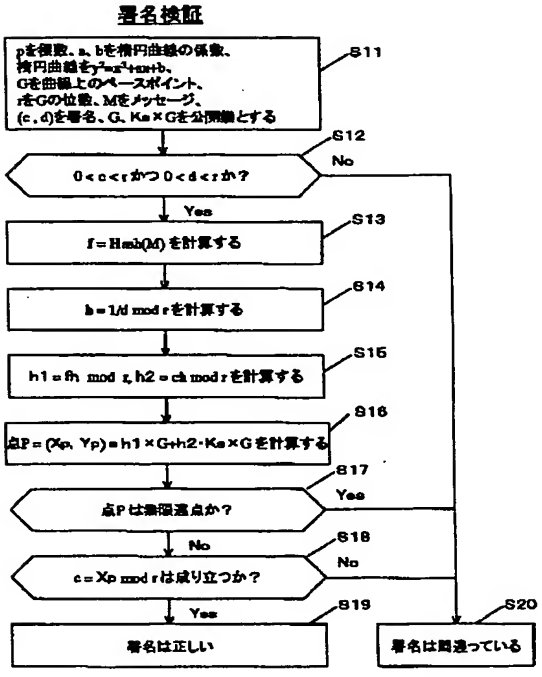


署名生成(IEEE P1363/D13)

【図９】

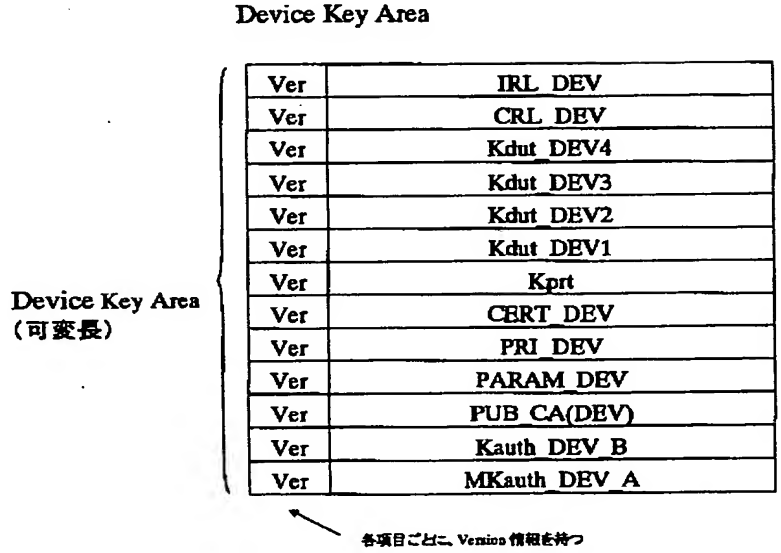


【図１３】

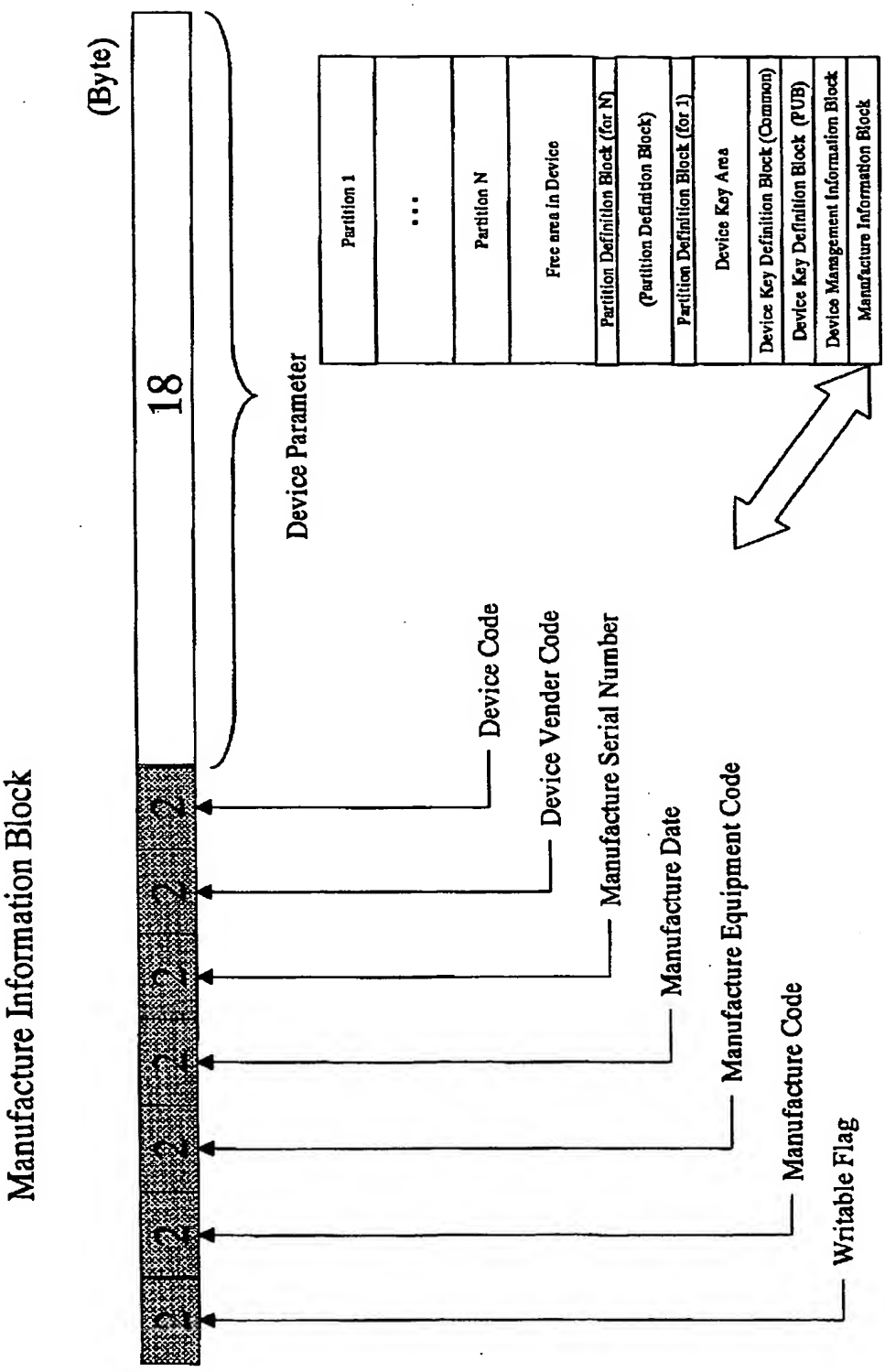


署名検証(IEEE P1363/D13)

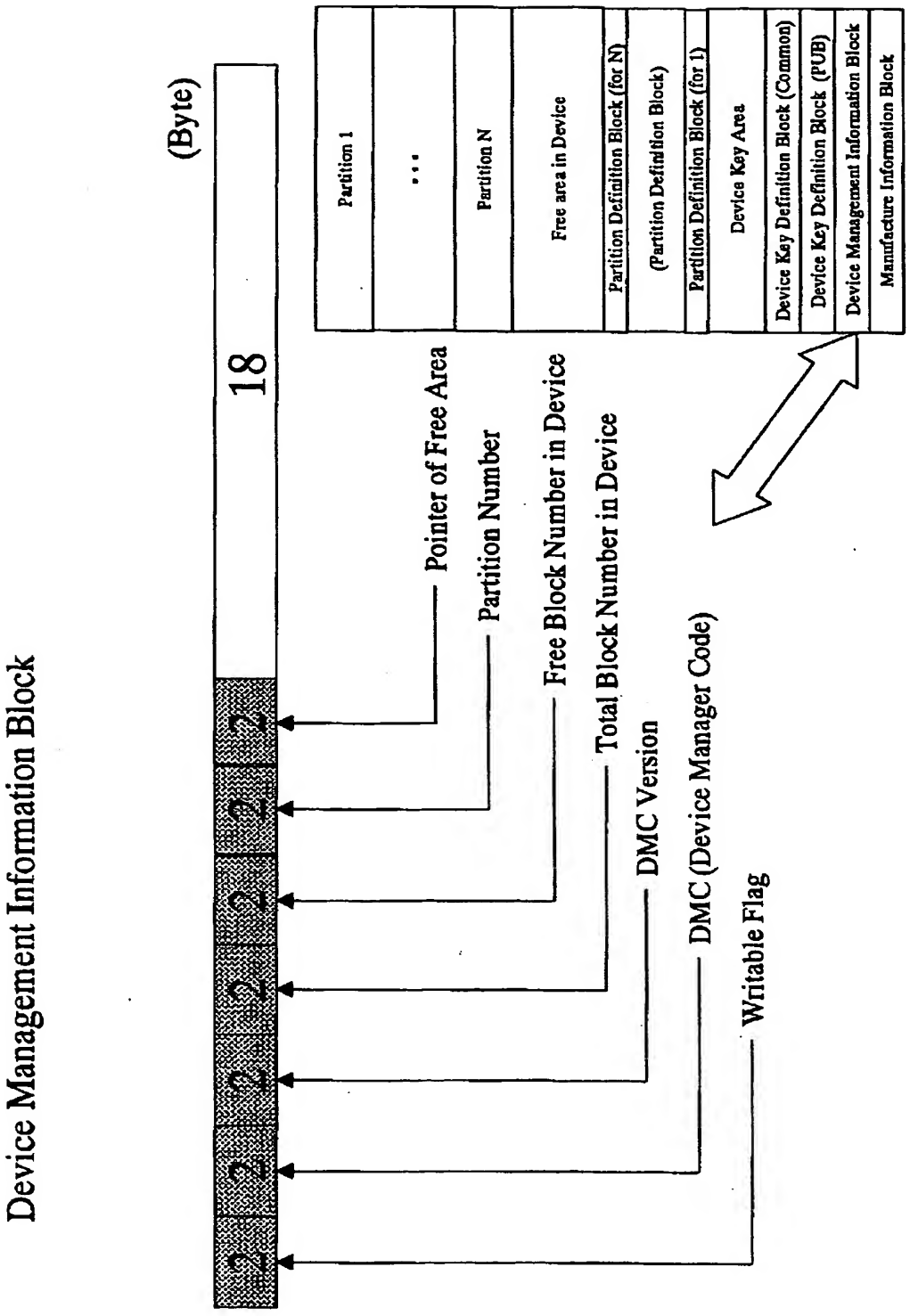
【図１８】



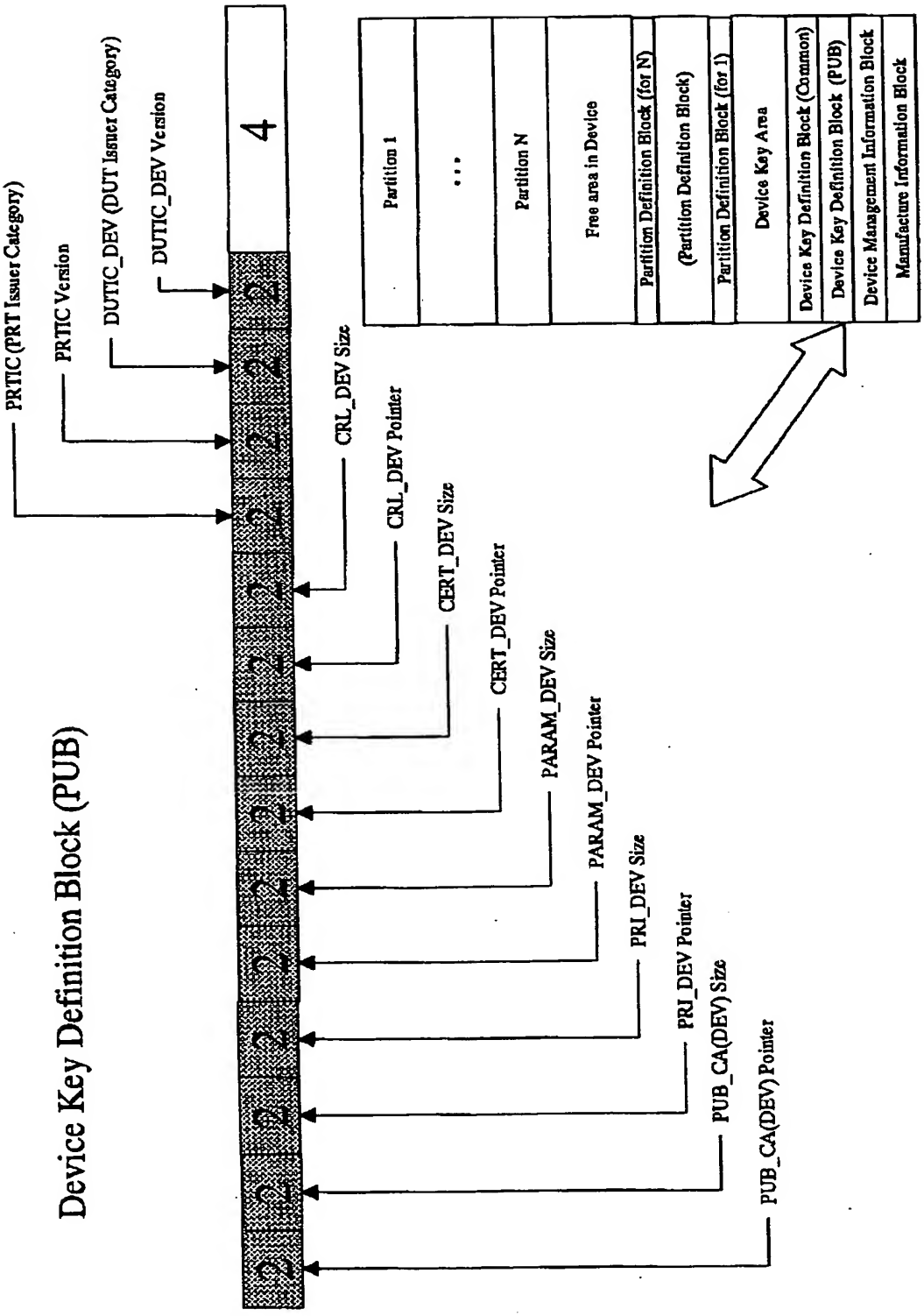
【図１４】


Manufacture Information Block
(Byte)
18
2
2
2
2
2
2
2
2
Device Parameter
Partition 1
...
Partition N
Free area in Device
Partition Definition Block (for N)
(Partition Defination Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (FUB)
Device Management Information Block
Manufacture Information Block
Device Code
Device Vender Code
Manufacture Serial Number
Manufacture Date
Manufacture Equipment Code
Manufacture Code
Writable Flag

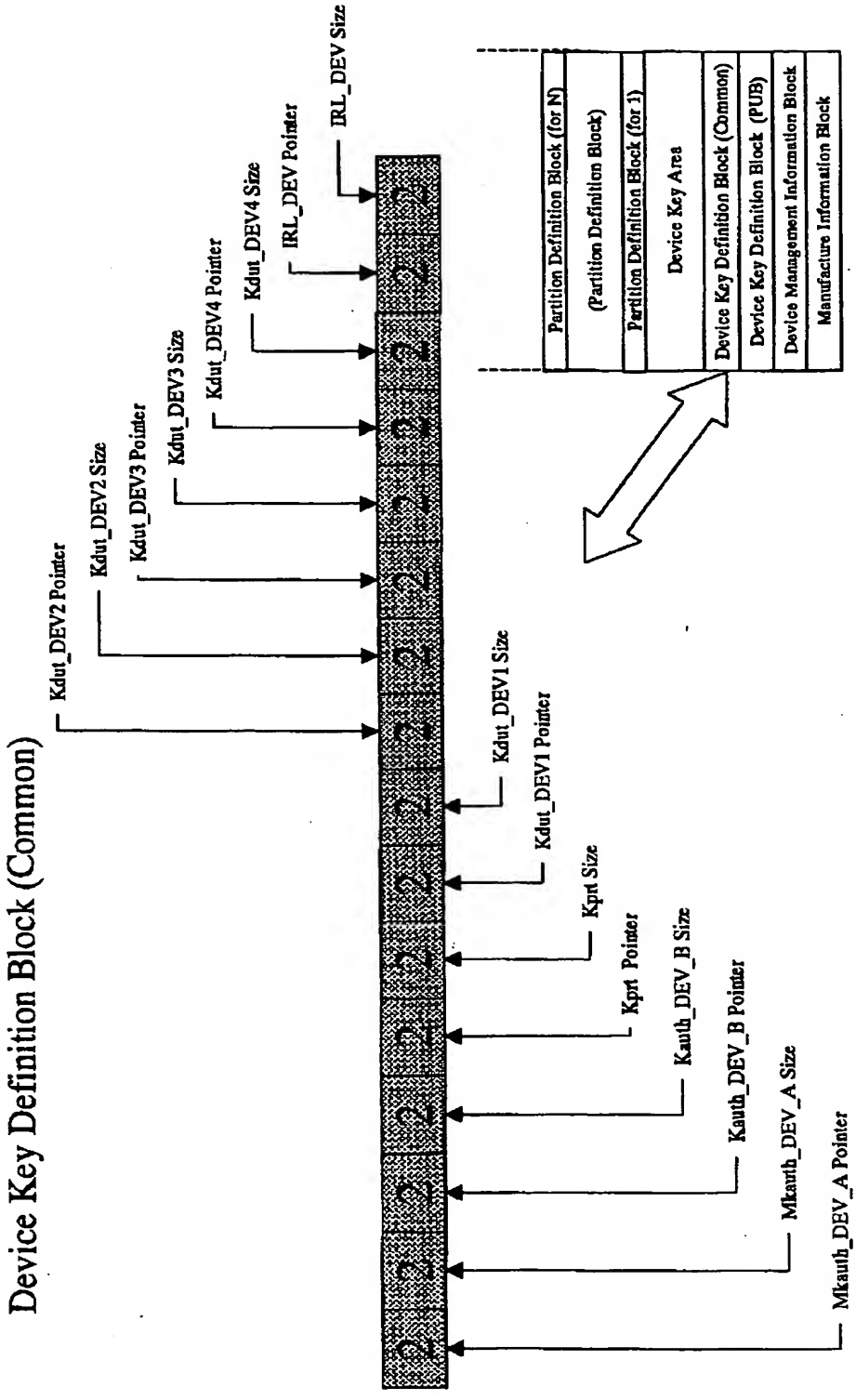
【図１５】


Device Management Information Block
(Byte)
18
2
2
2
2
2
2
2
2
2
Pointer of Free Area
Partition Number
Free Block Number in Device
Total Block Number in Device
DMC Version
DMC (Device Manager Code)
Writable Flag
Partition 1
...
Partition N
Free area in Device
Partition Definition Block (for N)
(Partition Definition Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (PUB)
Device Management Information Block
Manufacture Information Block

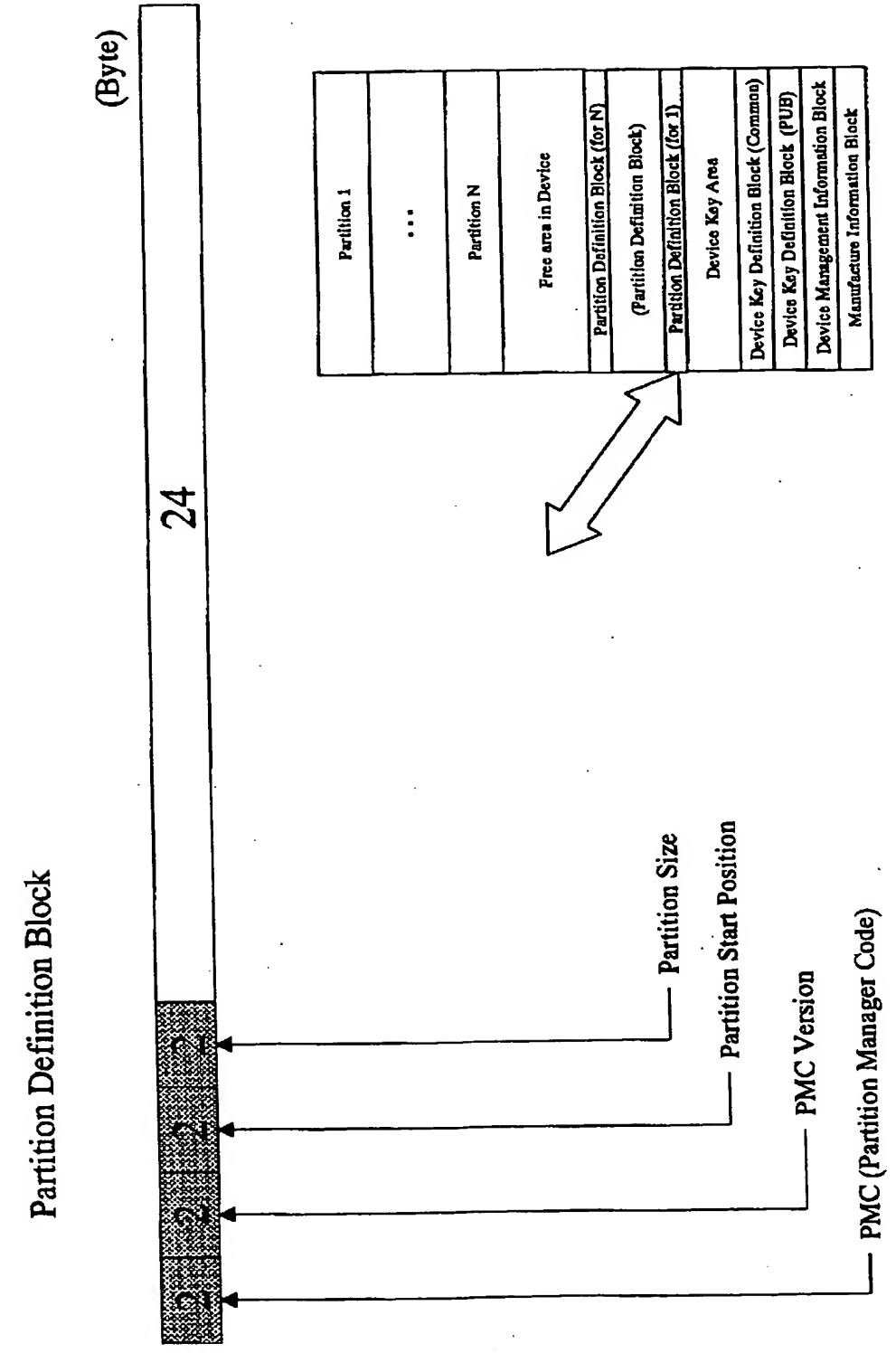
【図１６】


Device Key Definition Block (PUB)
PRTIC (PRT Issuer Category)
PRTIC Version
DUTIC_DEV (DUT Issuer Category)
DUTIC_DEV Version
CRL_DEV Size
CRL_DEV Pointer
CERT_DEV Size
CERT_DEV Pointer
PARAM_DEV Size
PARAM_DEV Pointer
PRI_DEV Size
PRI_DEV Pointer
PUB_CA(DEV) Size
PUB_CA(DEV) Pointer
Partition 1
...
Partition N
Free area in Device
Partition Definition Block (for N)
(Partition Definition Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (PUB)
Device Management Information Block
Manufacture Information Block

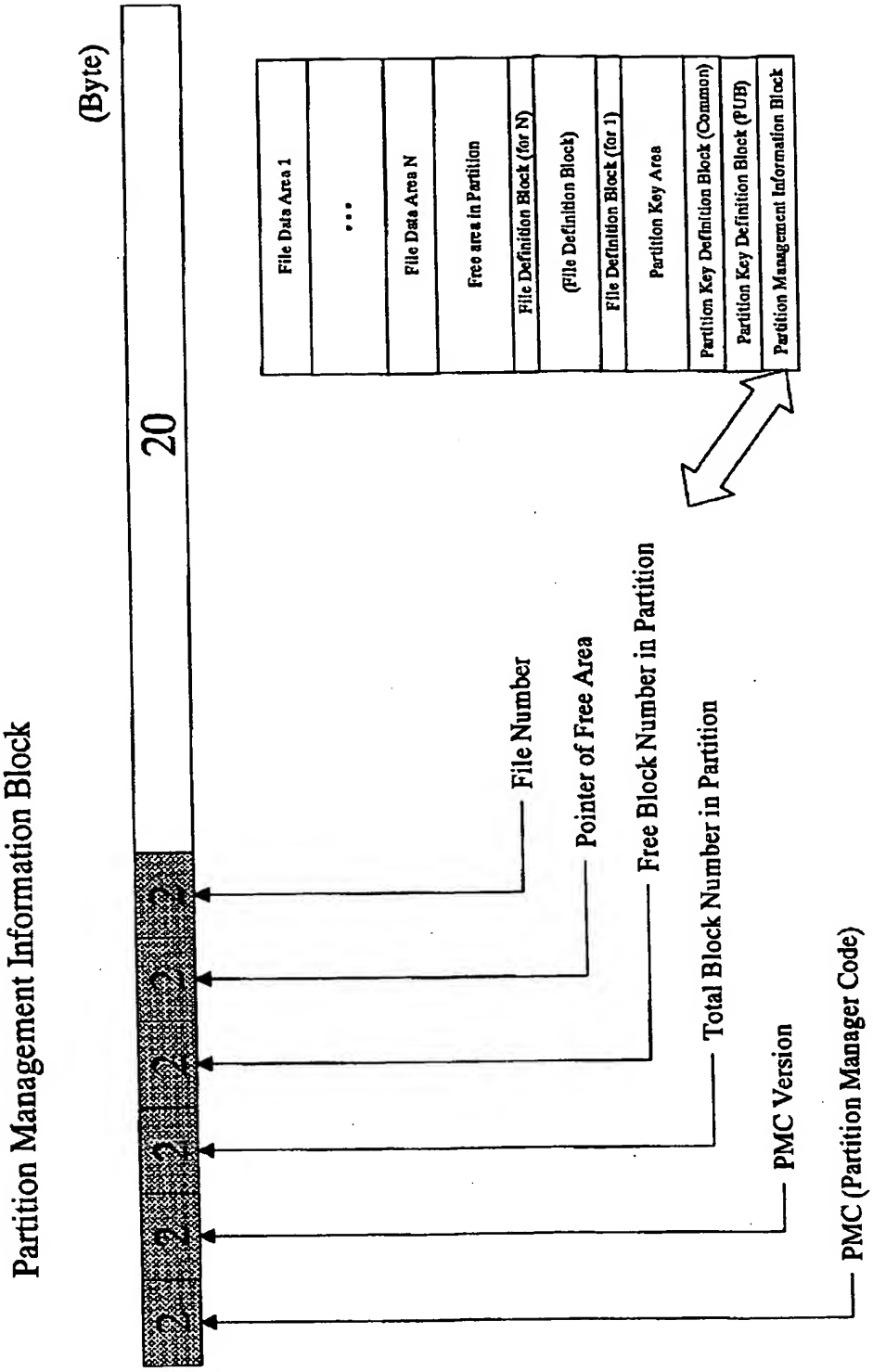
【図１７】


Device Key Definition Block (Common)
Kdut_DEV2 Pointer
Kdut_DEV2 Size
Kdut_DEV3 Pointer
Kdut_DEV3 Size
Kdut_DEV4 Pointer
Kdut_DEV4 Size
IRL_DEV Pointer
IRL_DEV Size
Kdut_DEV1 Pointer
Kdut_DEV1 Size
Kprt Pointer
Kprt Size
Kauth_DEV_B Pointer
Kauth_DEV_B Size
Mkauth_DEV_A Pointer
Mkauth_DEV_A Size
Partition Definition Block (for N)
(Partition Definition Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (PUB)
Device Management Information Block
Manufacture Information Block

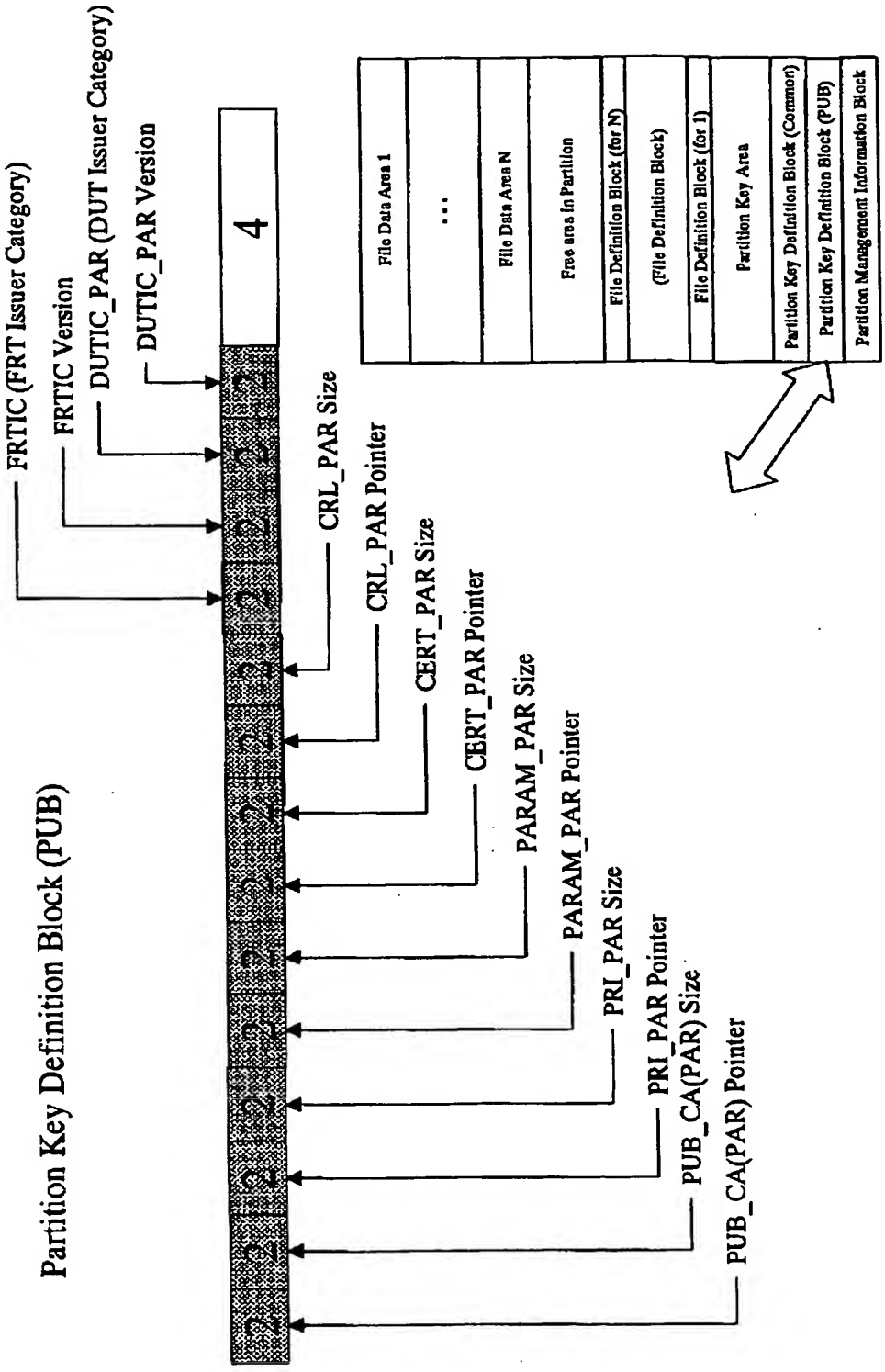
【図１９】


Partition Definition Block
(Byte)
24
Partition Size
Partition Start Position
PMC Version
PMC (Partition Manager Code)
Partition 1
...
Partition N
Free area in Device
Partition Definition Block (for N)
(Partition Definition Block)
Partition Definition Block (for 1)
Device Key Area
Device Key Definition Block (Common)
Device Key Definition Block (PUB)
Device Management Information Block
Manufacture Information Block

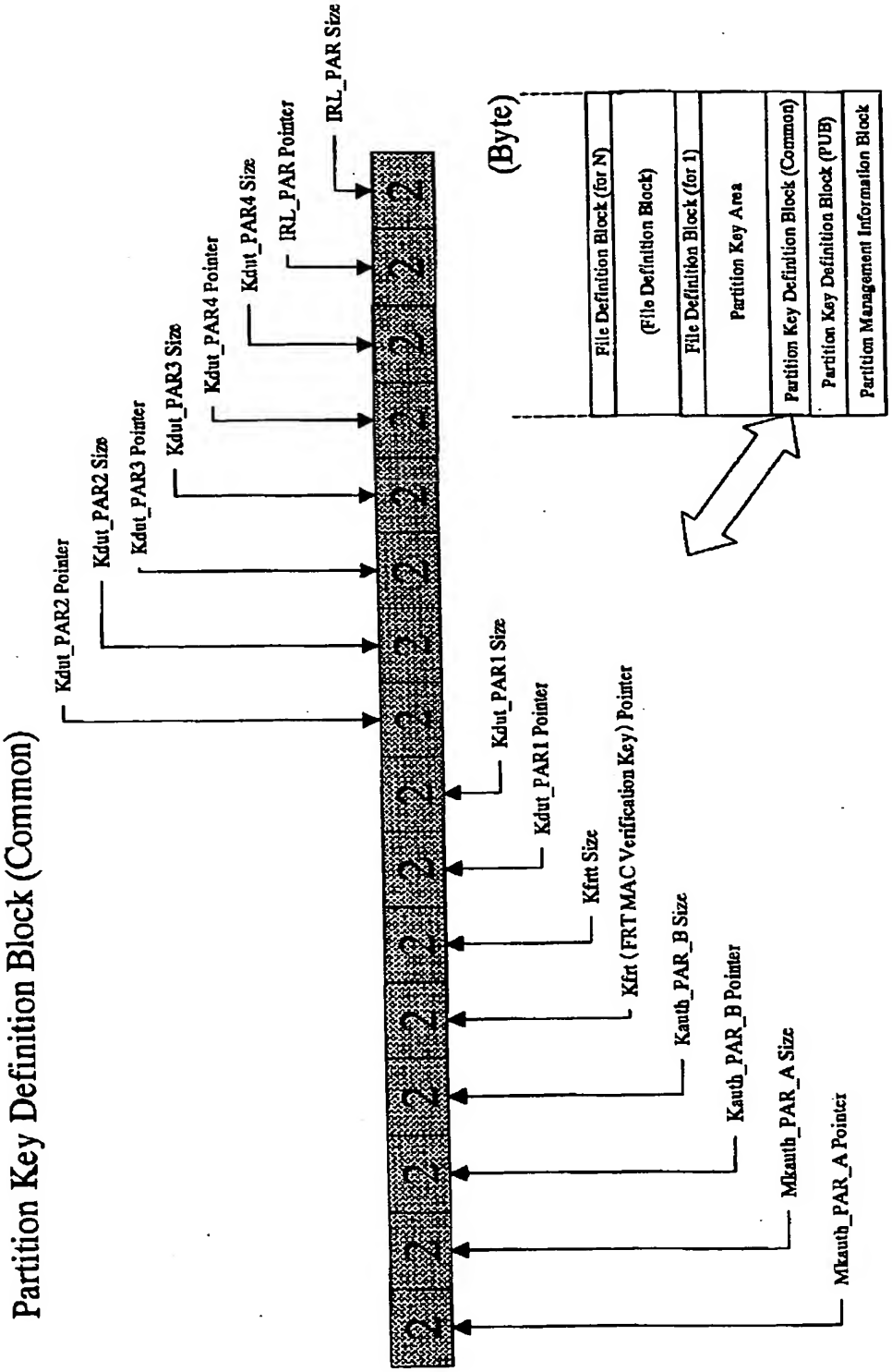
【図２０】


Partition Management Information Block
(Byte)
20
2
2
2
2
2
2
2
2
File Number
Pointer of Free Area
Free Block Number in Partition
Total Block Number in Partition
PMC Version
PMC (Partition Manager Code)
File Data Area 1
...
File Data Area N
Free area in Partition
File Definition Block (for N)
(File Definition Block)
File Definition Block (for 1)
Partition Key Area
Partition Key Definition Block (Common)
Partition Key Definition Block (PUB)
Partition Management Information Block

【図２１】


Partition Key Definition Block (PUB)
FRTIC (FRT Issuer Category)
FRTIC Version
DUTIC_PAR (DUT Issuer Category)
DUTIC_PAR Version
CRL_PAR Size
CRL_PAR Pointer
CERT_PAR Size
CERT_PAR Pointer
PARAM_PAR Size
PARAM_PAR Pointer
PRI_PAR Size
PRI_PAR Pointer
PUB_CA(PAR) Size
PUB_CA(PAR) Pointer
File Data Area 1
...
File Data Area N
Free area in Partition
File Definition Block (for N)
(File Definition Block)
File Definition Block (for 1)
Partition Key Area
Partition Key Definition Block (Common)
Partition Key Definition Block (PUB)
Partition Management Information Block

【図２２】


Partition Key Definition Block (Common)
Kdut_PAR2 Pointer
Kdut_PAR2 Size
Kdut_PAR3 Pointer
Kdut_PAR3 Size
Kdut_PAR4 Pointer
Kdut_PAR4 Size
IRL_PAR Pointer
IRL_PAR Size
Kdut_PAR1 Size
Kdut_PAR1 Pointer
Kfrt Size
Kfrt (FRT MAC Verification Key) Pointer
Kauth_PAR_B Size
Kauth_PAR_B Pointer
Mkauth_PAR_A Size
Mkauth_PAR_A Pointer
(Byte)
File Definition Block (for N)
(File Definition Block)
File Definition Block (for 1)
Partition Key Area
Partition Key Definition Block (Common)
Partition Key Definition Block (PUB)
Partition Management Information Block

【図２３】

Partition Key Area

Partition Key Area
（可変長）

Ver	IRL_PAR
Ver	CRL_PAR
Ver	Kdut_PAR4
Ver	Kdut_PAR3
Ver	Kdut_PAR2
Ver	Kdut_PAR1
Ver	Kfrt
Ver	CERT_PAR
Ver	PRI_PAR
Ver	PARAM_PAR
Ver	PUB_CA(PAR)
Ver	Kauth_PAR_B
Ver	MKauth_PAR_A

各項目ごとに、Version 情報を持つ

【図２５】

File Structure

Type Code	Structure Type
0001	Random
0002	Purse
0003	Cyclic
0004	Log
0005	Key
0006	複合ファイル1
0007	複合ファイル2

【図２６】

Partition Registration Ticket (PRT) Format

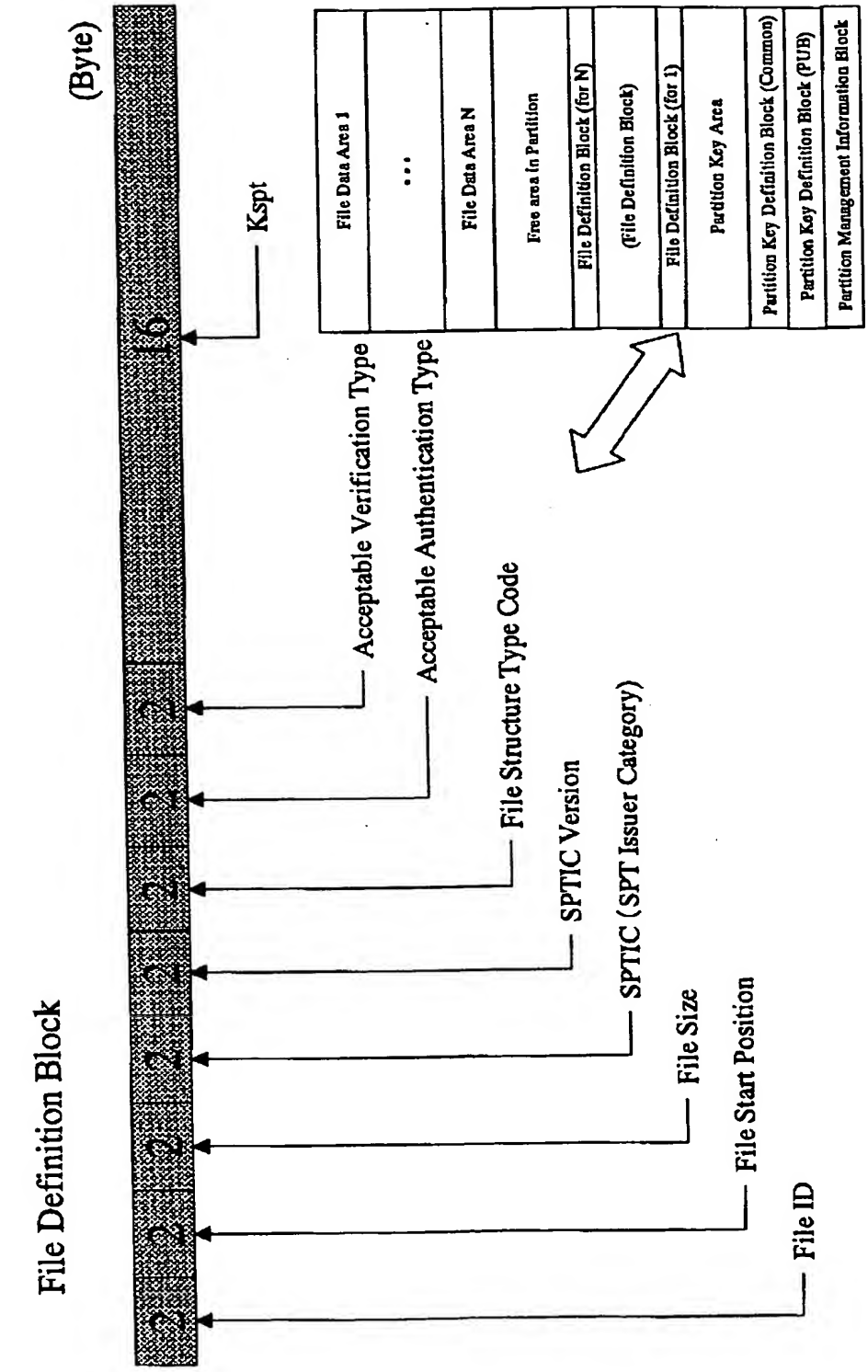
Ticket Type (=PRT)
Format Version
Ticket Issuer (=DMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属
Authentication Type
Ticket User の識別子
PMC
PMC Version
Operation Type
Partition Size
Integrity Check Type
Integrity Check Value

【図２７】

File Registration Ticket (FRT) Format

Ticket Type (=FRT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
SPTIC
SPTIC Ver
File ID
Operation Type
File Size
File Structure Type
Acceptable Authentication Type
Kspt_Encrypted
Integrity Check Type
Integrity Check Value

【図２４】


File Definition Block
(Byte)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Kspt
Acceptable Verification Type
Acceptable Authentication Type
File Structure Type Code
SPTIC Version
SPTIC (SPT Issuer Category)
File Size
File Start Position
File ID
File Data Area 1
...
File Data Area N
Free area in Partition
File Definition Block (for N)
(File Definition Block)
File Definition Block (for 1)
Partition Key Area
Partition Key Definition Block (Common)
Partition Key Definition Block (PUB)
Partition Management Information Block

【図２８】

Service Permission Ticket (SPT) Format （1）

Ticket Type (=SPT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
File ID
File Access Mode
Integrity Check Type
Integrity Check Value

} ペア

【図２９】

File Access Mode

Access Mode Code	Access Mode	Access Mode Code	Access Mode	Access Mode Code	Access Mode
0001	Read	0011	Erase	0021	Certificate 登録
0002	Write	0012	Add	0022	Certificate 検証
0003	暗号化 Read	0013	Sub	0023	鍵ペア生成
0004	暗号化 Write	0014	Compare	0024	鍵ペア検証
0005	MAC付 Read	0015	暗号化	0025	相互認証
0006	MAC付 Write	0016	復号	0026	入金系
0007	MAC付 暗号化 Read	0017	署名生成	0027	出金系
0008	MAC付 暗号化 Write	0018	署名検証	0028	
0009	Write Only	0019	MAC生成	0029	
0010	Write Once	0020	MAC検証	0030	

【図５１】

Device 内認証テーブル

Group	公開鍵方式認証者情報	Session Key	共通鍵方式認証者情報	Session Key
DMC	DN, Serial Number, Category	Kses1	—	—
PMC1	—	—	ID_RW	Kses2
PMC 2	DN, Serial Number, Category	Kses3	ID_RW	Kses4

【図３０】

File Structure, Access Mode & Command

File Structure	Access Mode	Acceptable Command	File Structure	Access Mode	Acceptable Command
Random	Read	Read	複合ファイル（電子マネー）	入金系	Deposit
	Write	Write		出金系	Withdraw Make Receipt Read Receipt
	暗号化 Read	EncRead			
	暗号化 Write	EncWrite			
	MAC付 Read	MacRead			
	MAC付 Write	MacWrite			
	MAC付 暗号化Read	EncMacRead			
	MAC付 暗号化 Write	EncMacWrite			

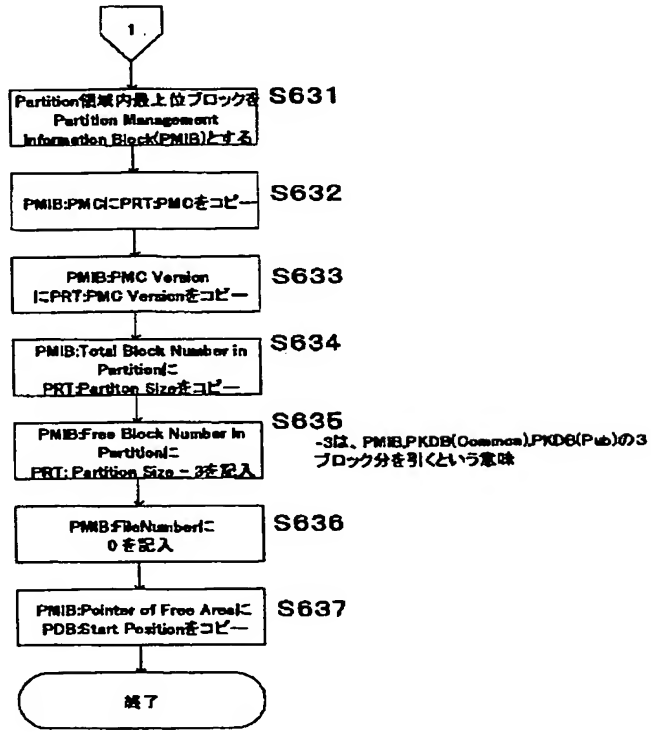
【図３１】

Service Permission Ticket (SPT) Format （2）

Ticket Type (=SPT)
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Authentication Flag
Ticket User の所属 (Group)
Authentication Type
Ticket User の識別子
File ID
File Access Mode
Group of Target File
Target File ID
Read/Write Permission
Integrity Check Type
Integrity Check Value

1セット

【図６１】



Partition作成・削除操作（続き）

【図32】

Data Update Ticket (DUT) Format

Ticket Type (=DUT(DEV))
Format Version
Ticket Issuer (=DMC)
Serial Number
Size of Ticket
Ticket User の所属
Ticket User の識別子
Authentication Type
Encrypted Flag
Old DataCode
Data Version Rule
Data Version Condition
Size of New Data
New Data
New Data Version
Integrity Check Type
Integrity Check Value

DUT(DEV)

Ticket Type (=DUT(PAR))
Format Version
Ticket Issuer (=PMC)
Serial Number
Size of Ticket
Ticket User の所属
Ticket User の識別子
Authentication Type
Encrypted Flag
Old Data Code
Data Version Rule
Data Version Condition
Size of New Data
New Data
New Data Version
Integrity Check Type
Integrity Check Value

DUT(PAR)

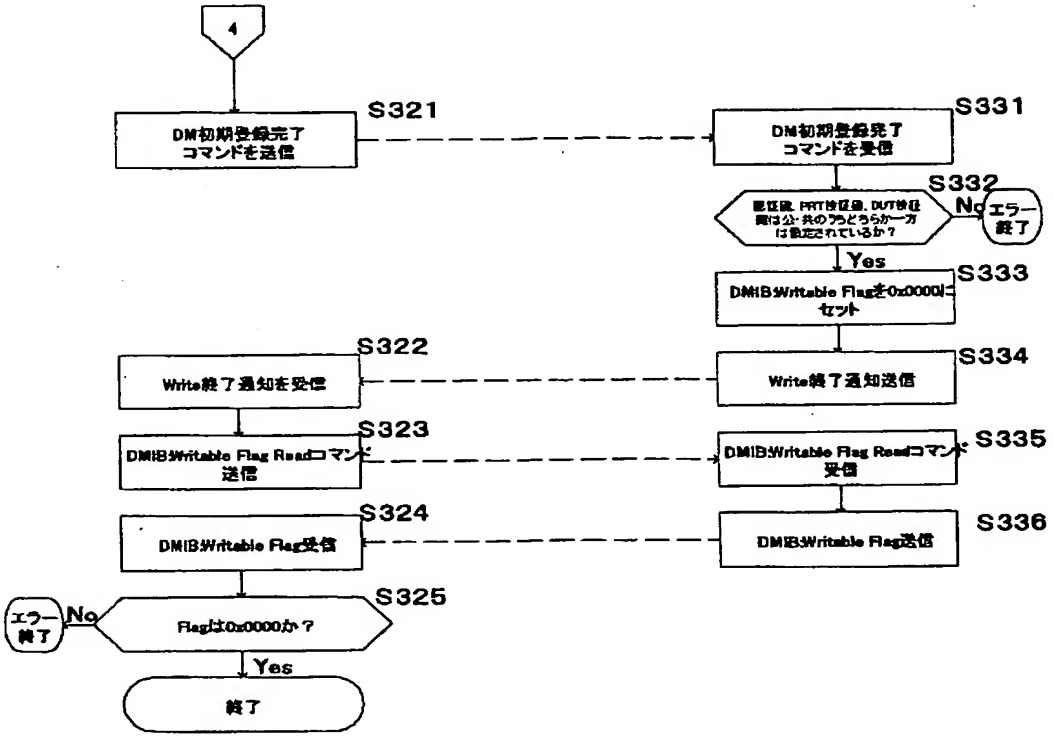
特開平14-278838

【図３３】

New Data Update Code

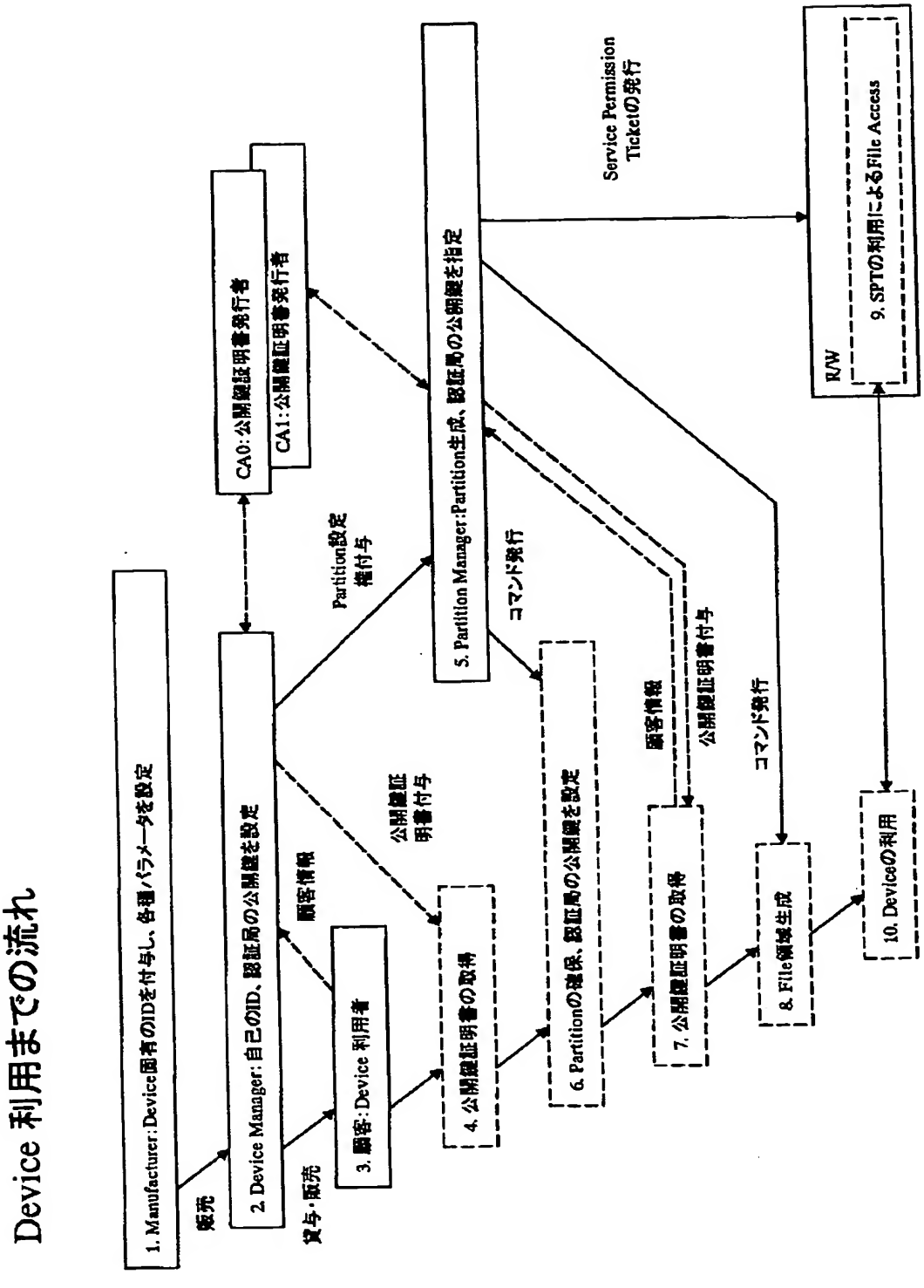
Code	New Data	Code	New Data	Code	New Data
0001	DMC	0011	DUTIC_PAR	0021	MKauth_DEV_A
0002	DMC Version	0012	DUTIC_PAR Version	0022	Kauth_DEV_B
0003	PMC	0013	Kdut_DEV1,2	0023	IRL_DEV
0004	PMC Version	0014	Kdut_PAR1,2	0024	IRL_PAR
0005	PRTIC	0015	Kprt	0025	
0006	PRTIC Version	0016	Kprt Version	0026	
0007	FRTIC	0017	Kfrt	0027	
0008	FRTIC Version	0018	Kfrt Version	0028	
0009	DUTIC_DEV	0019	Kspt	0029	
0010	DUTIC_DEV Version	0020	Kspt Version	0030	

【図４０】

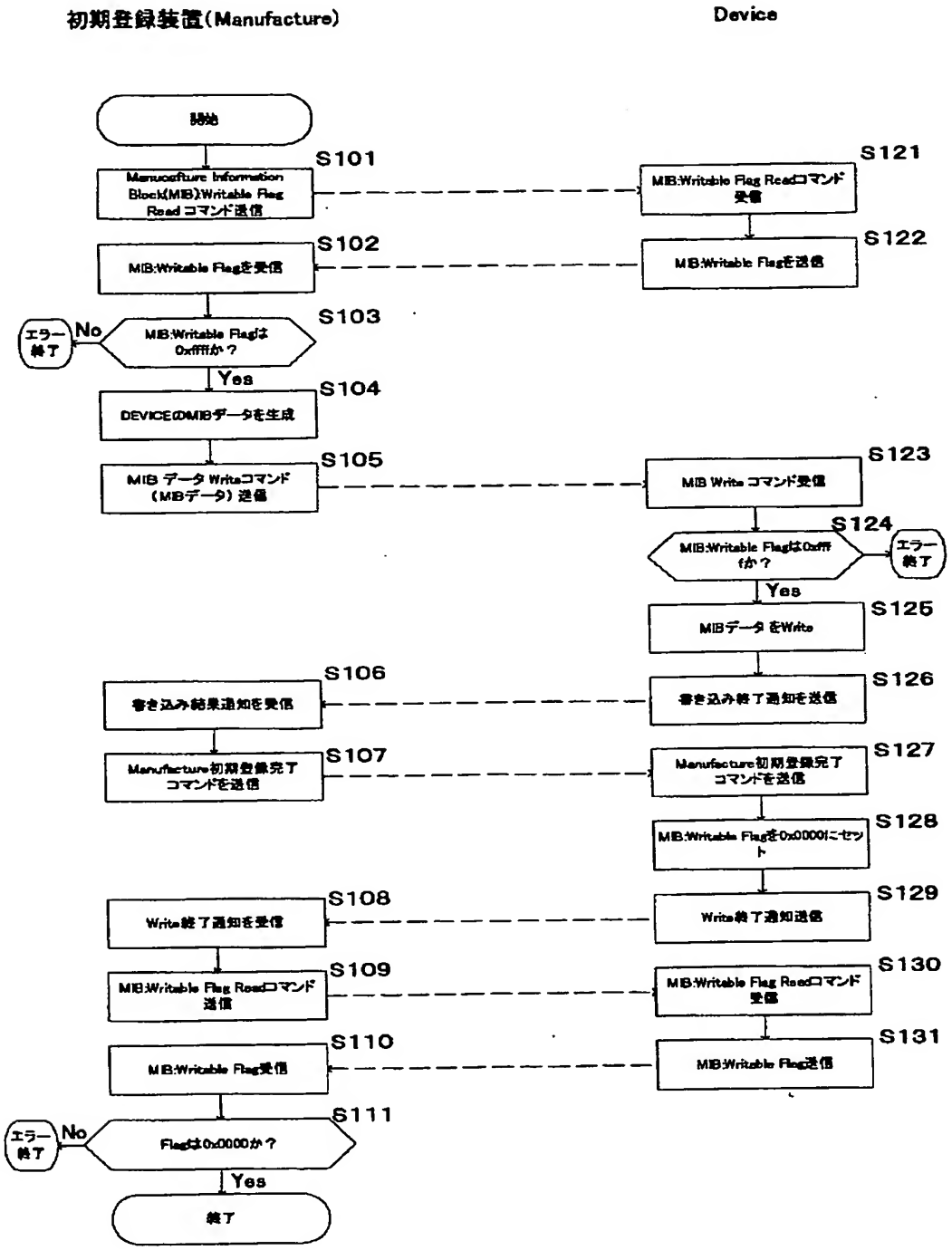


Device初期登録(DM)(続き)

【図３４】

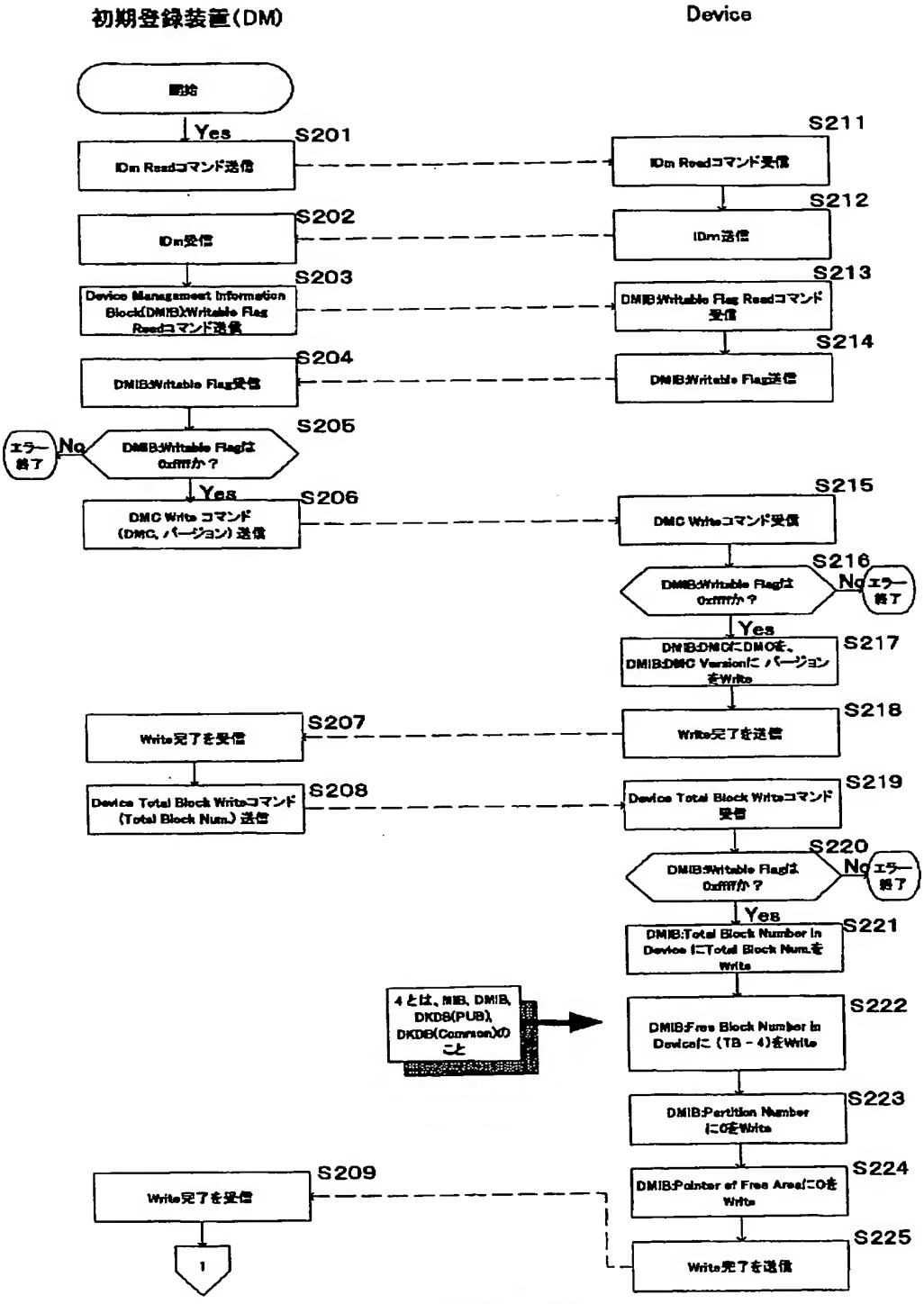

Device 利用までの流れ
1. Manufacturer:Device固有のIDを付与し、各種パラメータを設定
販売
2. Device Manager:自己のID、認証局の公開鍵を設定
貸与・販売
3. 顧客:Device 利用者
顧客情報
4. 公開鍵証明書の取得
公開鍵証明書付与
CA0:公開鍵証明書発行者
CA1:公開鍵証明書発行者
Partition設定権付与
5. Partition Manager:Partition生成、認証局の公開鍵を指定
コマンド発行
6. Partitionの確保、認証局の公開鍵を設定
顧客情報
7. 公開鍵証明書の取得
公開鍵証明書付与
コマンド発行
8. File領域生成
10. Deviceの利用
Service Permission Ticketの発行
R/W
9. SPTの利用によるFile Access

【図３５】



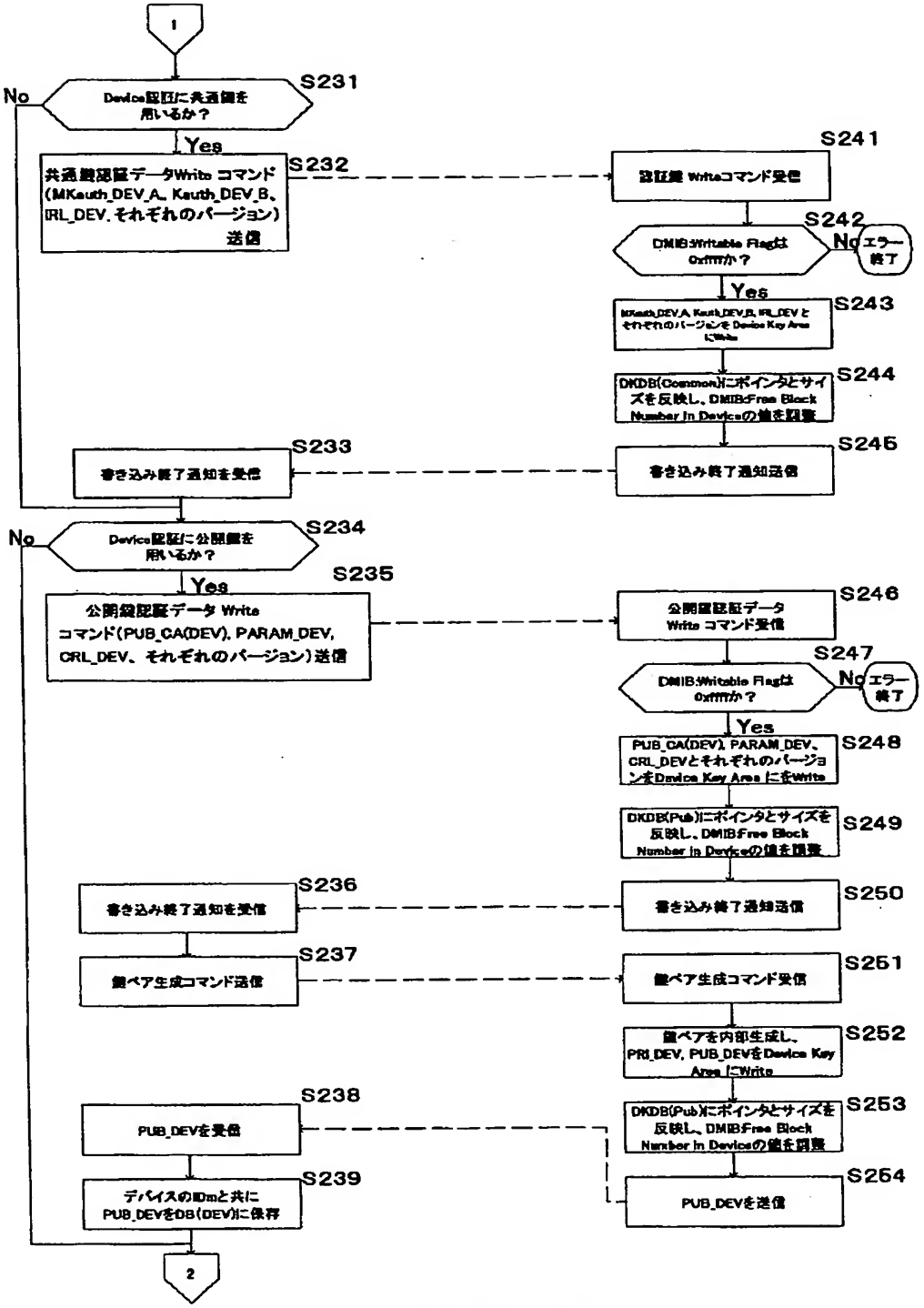
Device初期登録(Manufaoture)

【図３６】



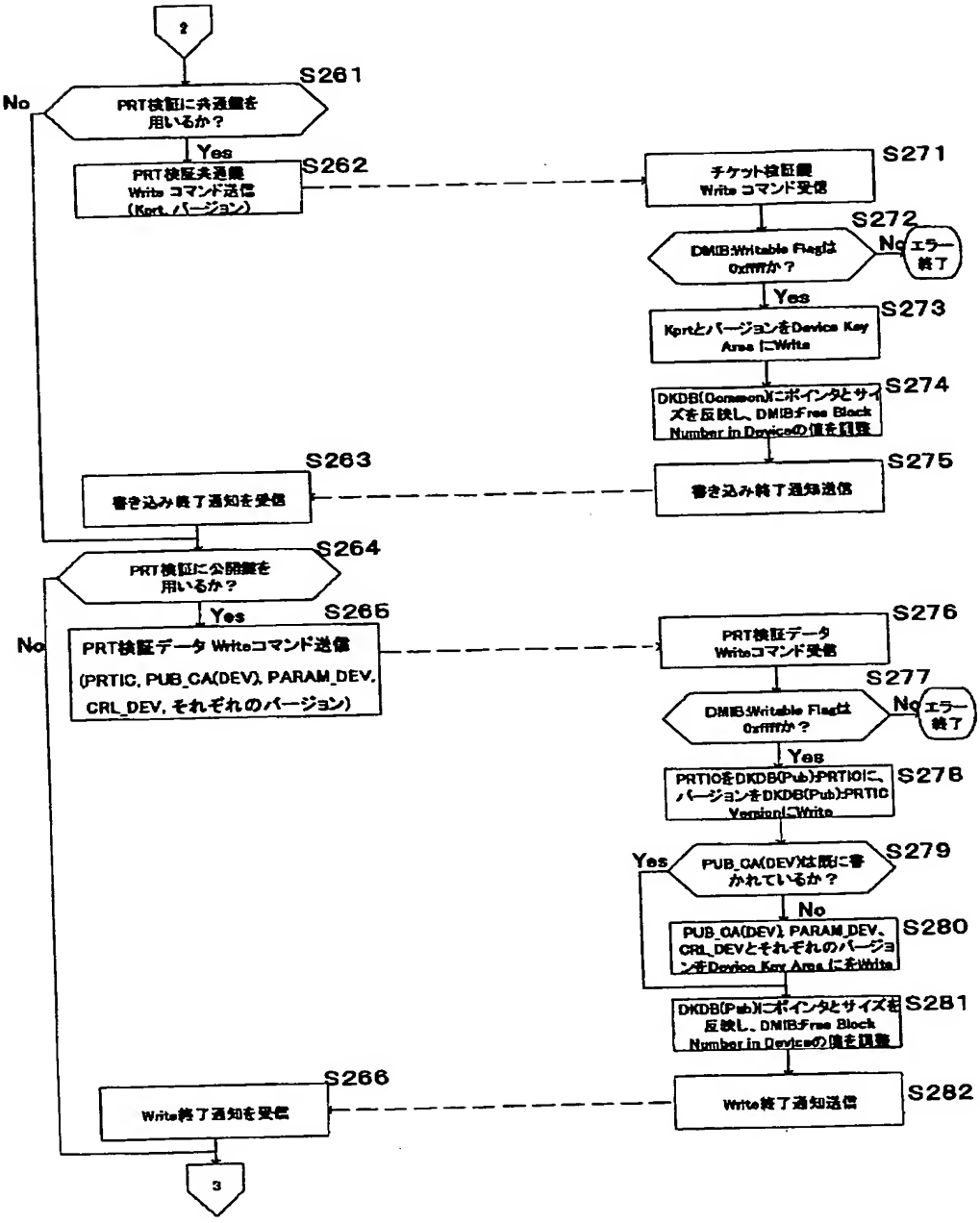
Decice初期登録(DM)

【図３７】



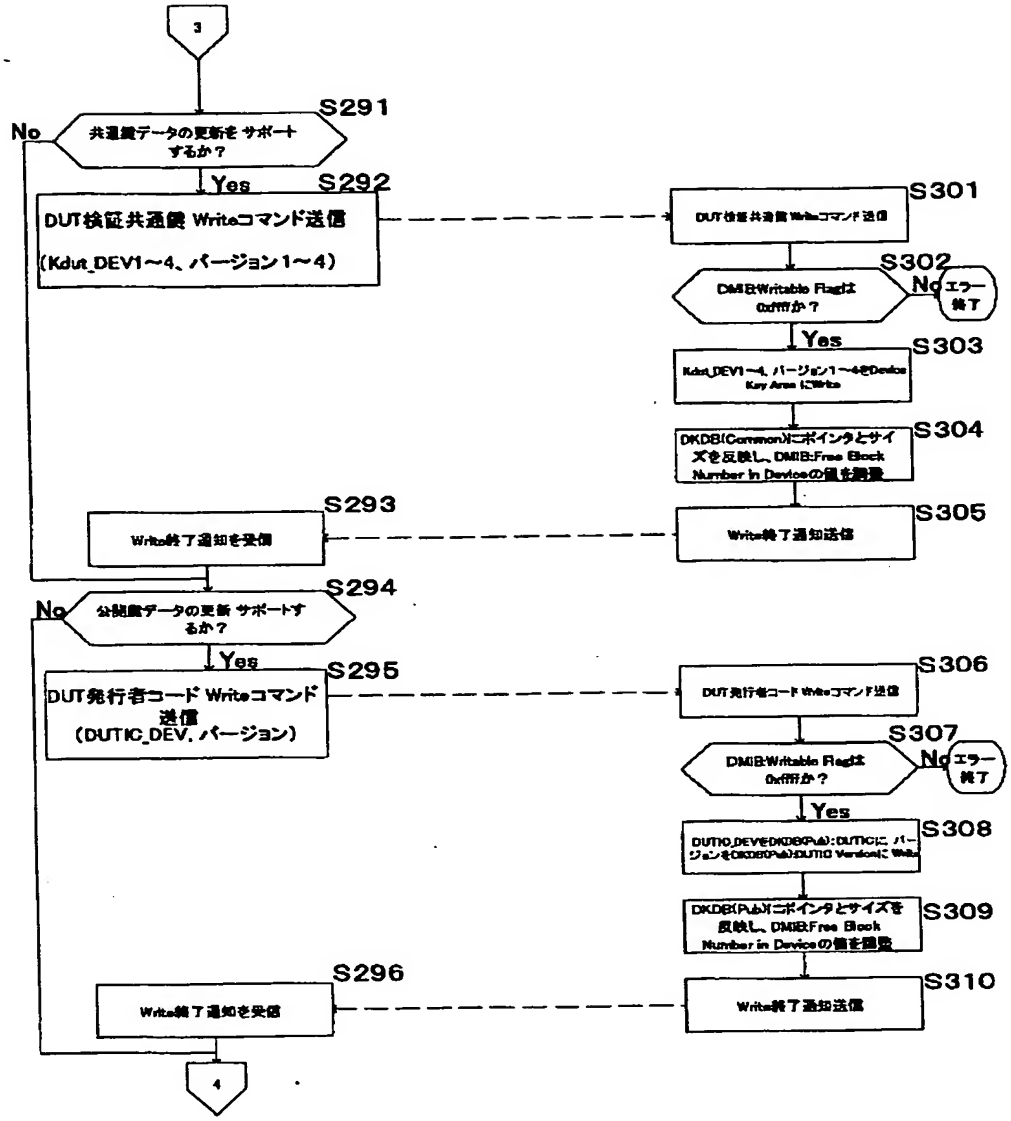
Device初期登録(DM)(続き)

【図３８】



Device初期登録(DM)(続き)

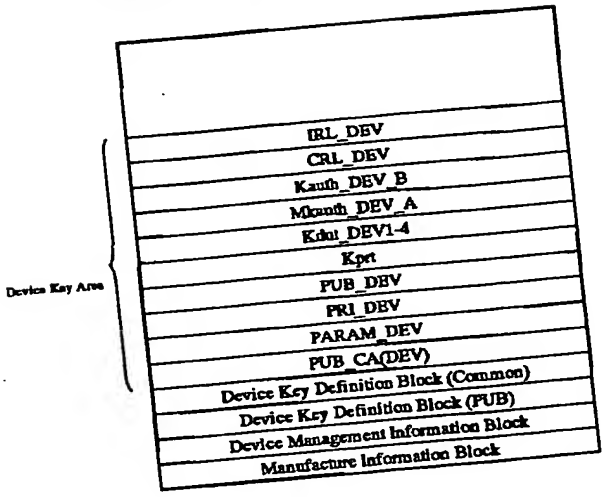
【図３９】



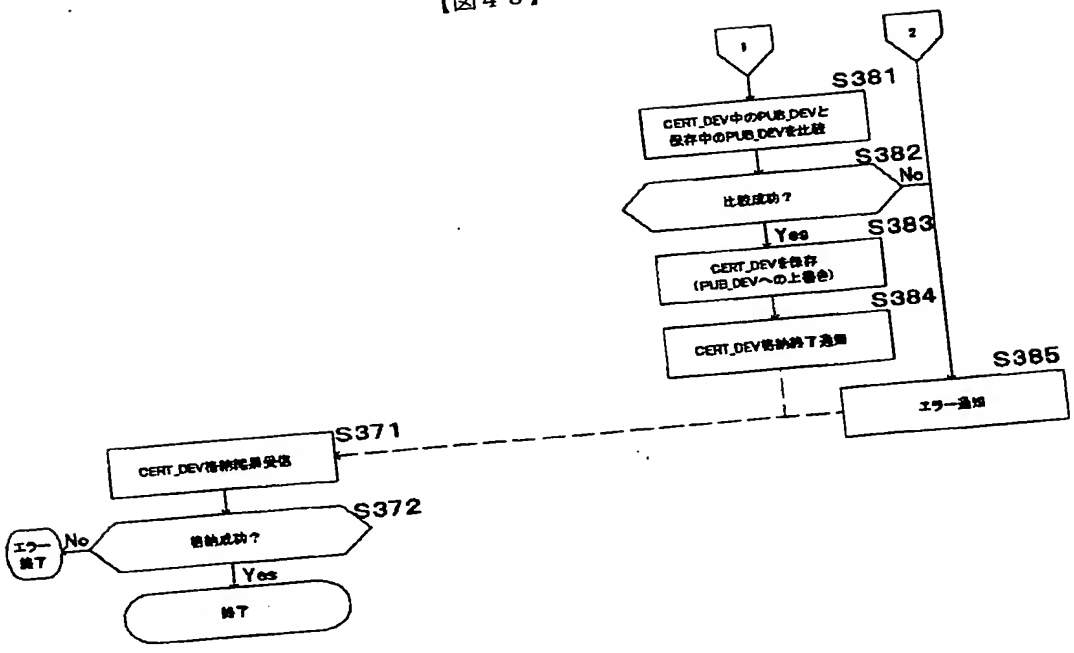
Device初期登録(DM)(続き)

【図４１】

初期データ書き込み

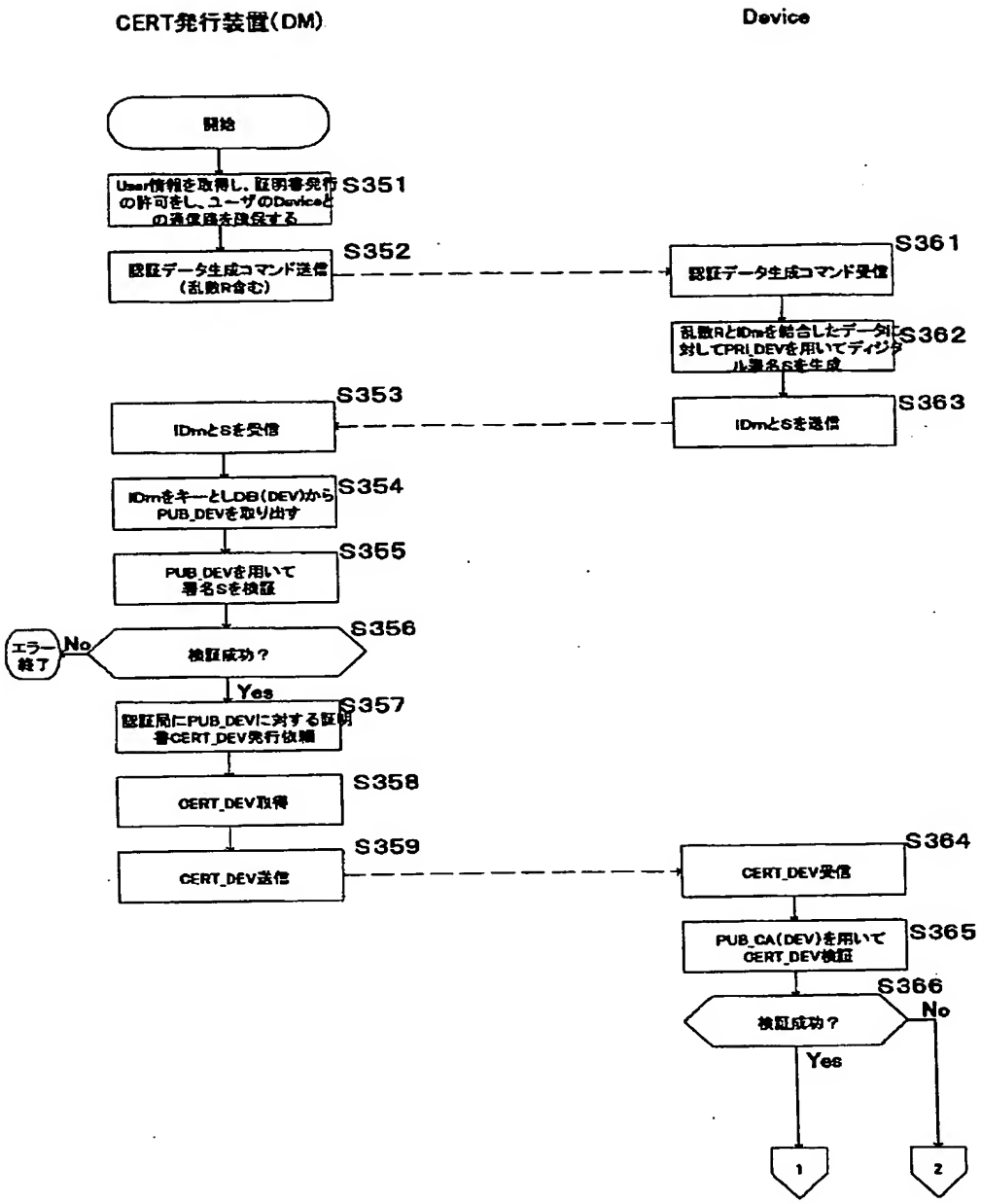


【図４３】



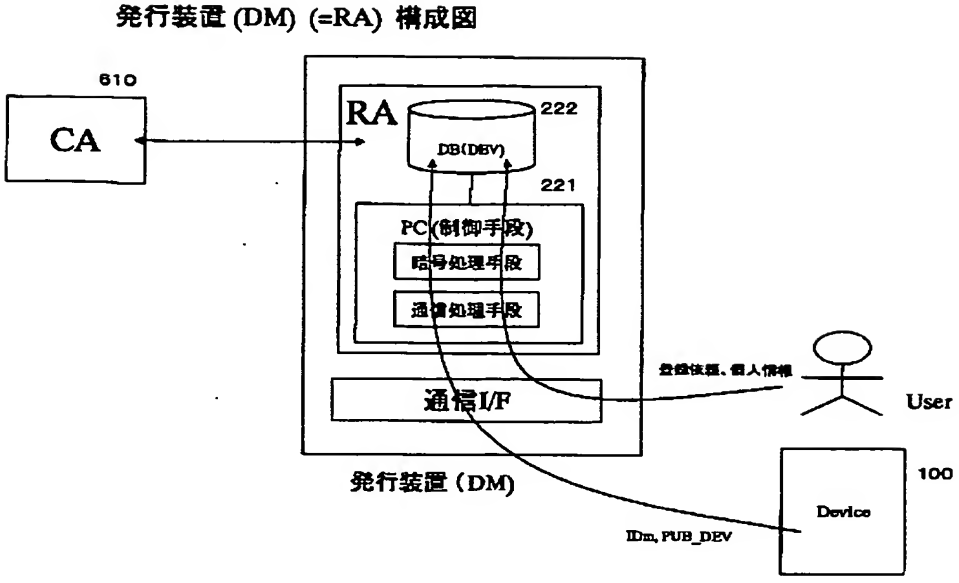
CERT発行(DM)(続き)

【図４２】

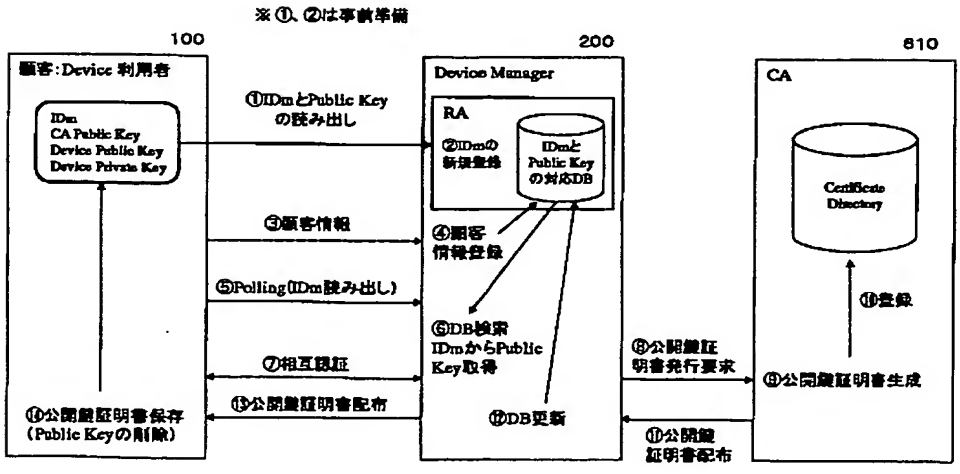


CERT発行(DM)

【図４４】

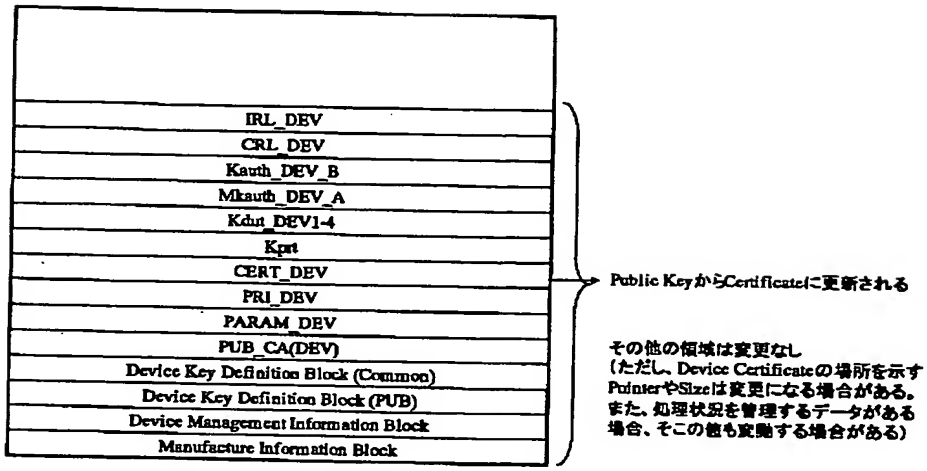

発行装置 (DM) (=RA) 構成図
610
CA
RA
222
DB(DEV)
221
PC(制御手段)
暗号処理手段
通信処理手段
通信I/F
発行装置 (DM)
User
100
Device
IDm, PUB_DEV

【図４５】


Certificate発行手順
※①、②は事前準備
100
顧客:Device 利用者
IDm
CA Public Key
Device Public Key
Device Private Key
⑭公開鍵証明書保存
(Public Keyの削除)
200
Device Manager
RA
②IDmの新規登録
IDmと Public Key の対応DB
610
CA
Certificate Directory
⑩登録
①IDmとPublic Key の読み出し
③顧客情報
④顧客情報登録
⑤Polling(IDm読み出し)
⑥DB検索 IDmからPublic Key取得
⑦相互認証
⑬公開鍵証明書配布
⑧公開鍵証明書発行要求
⑨公開鍵証明書生成
⑫DB更新
⑪公開鍵証明書配布

【図４６】

Certificate発行後の状態



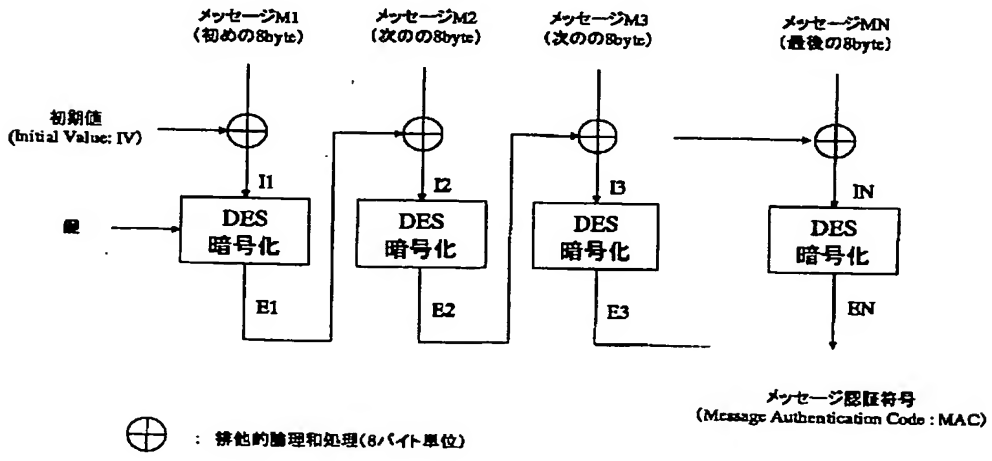
【図５２】

R/W 内認証テーブル

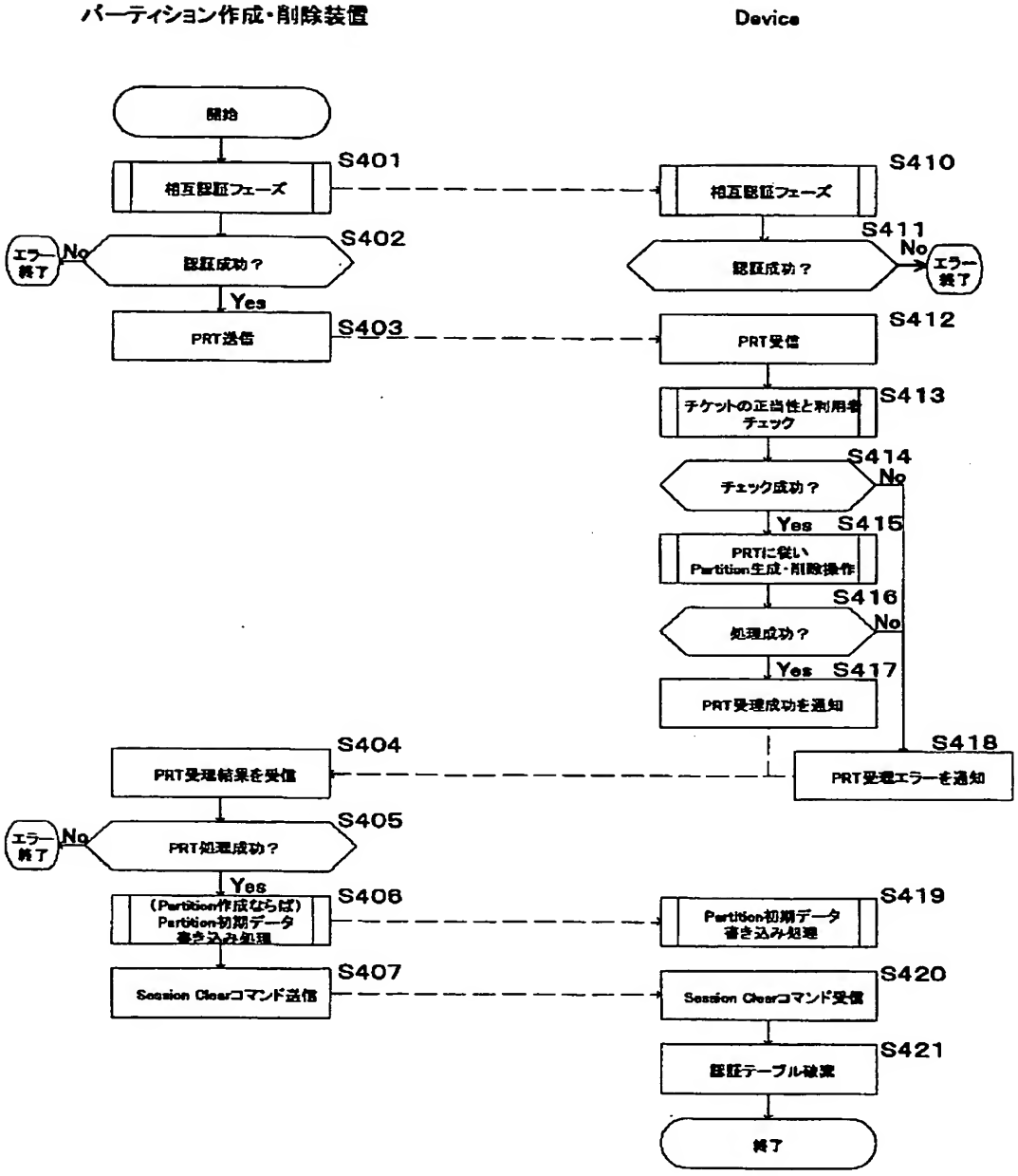
Group	公開鍵方式認証者情報	共通鍵方式認証者情報	Session Key
DMC	DN, Serial Number, Category	—	Kses1
PMC1	—	IDm	Kses2
PMC 2	DN, Serial Number, Category	IDm	Kses3

【図５９】

MAC生成方式

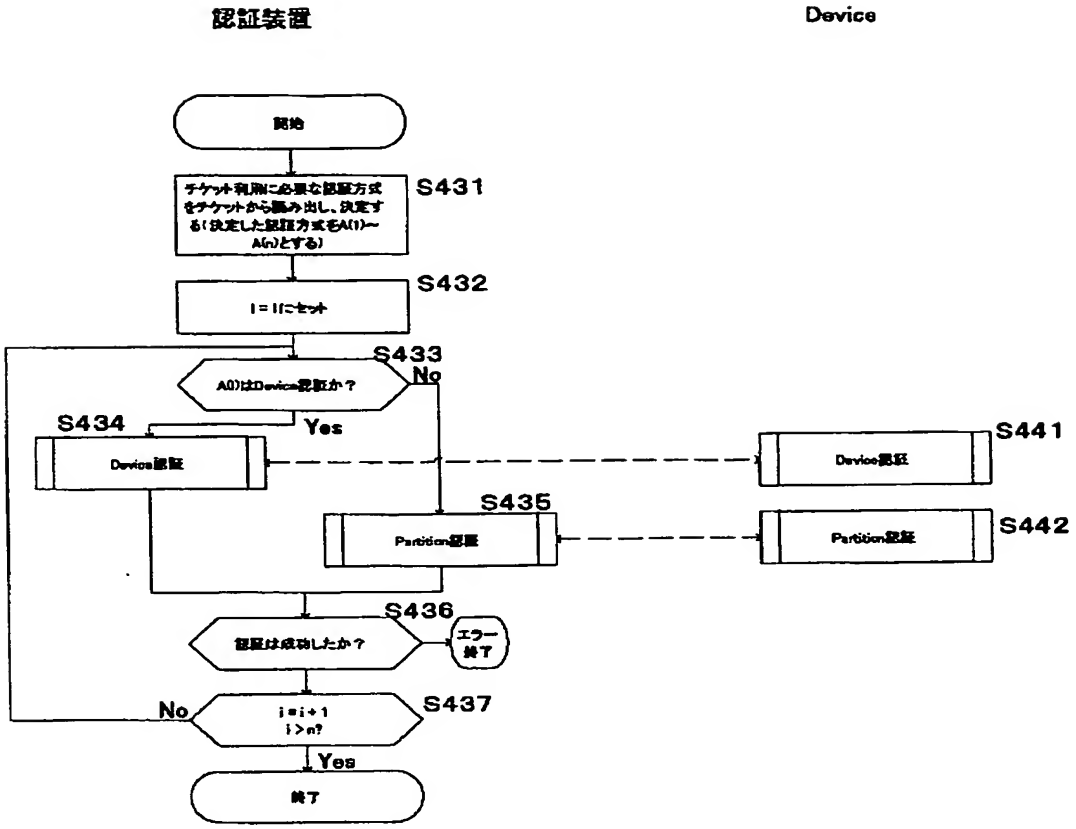


【図４７】



Partition生成・消去

【図４８】



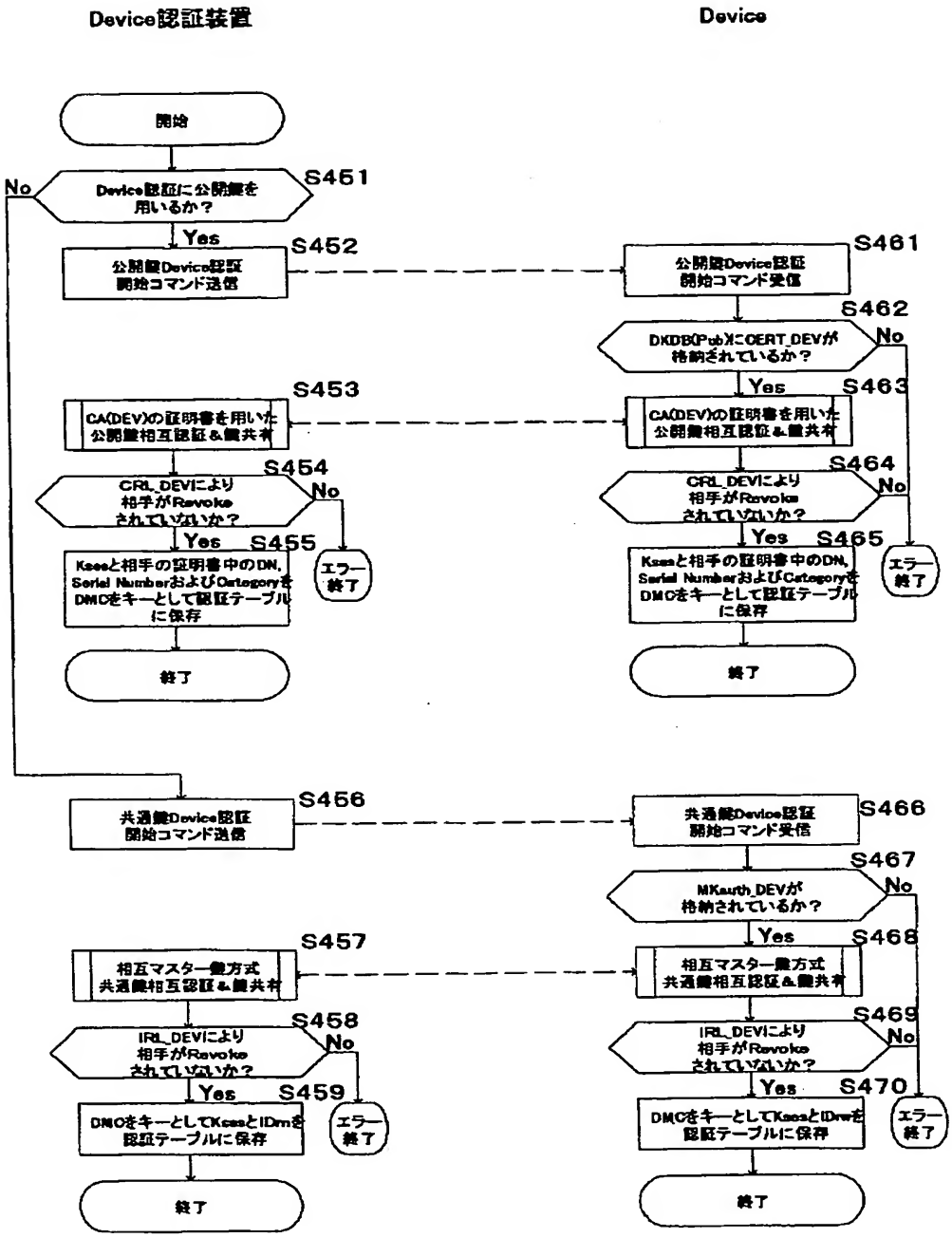
相互認証フェーズ

【図８２】

File Open テーブル(2)

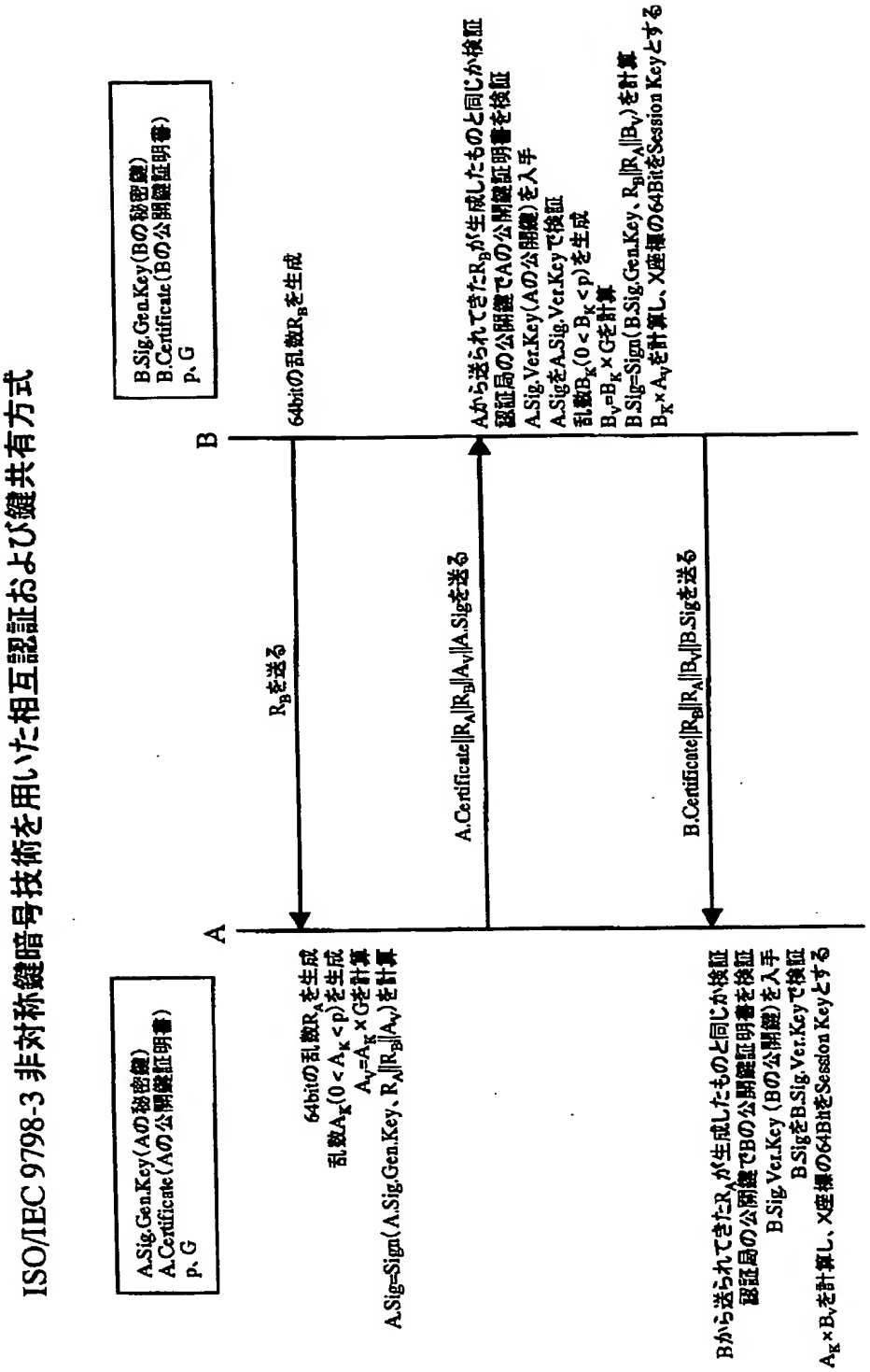
Group	File ID	File Access Mode	Group of Target File	Target File ID	Read/Write Permission
PMC1	0x0001	暗号化、復号			
PMC2	0x0002		PMC1	0x0001	Read

【図４９】

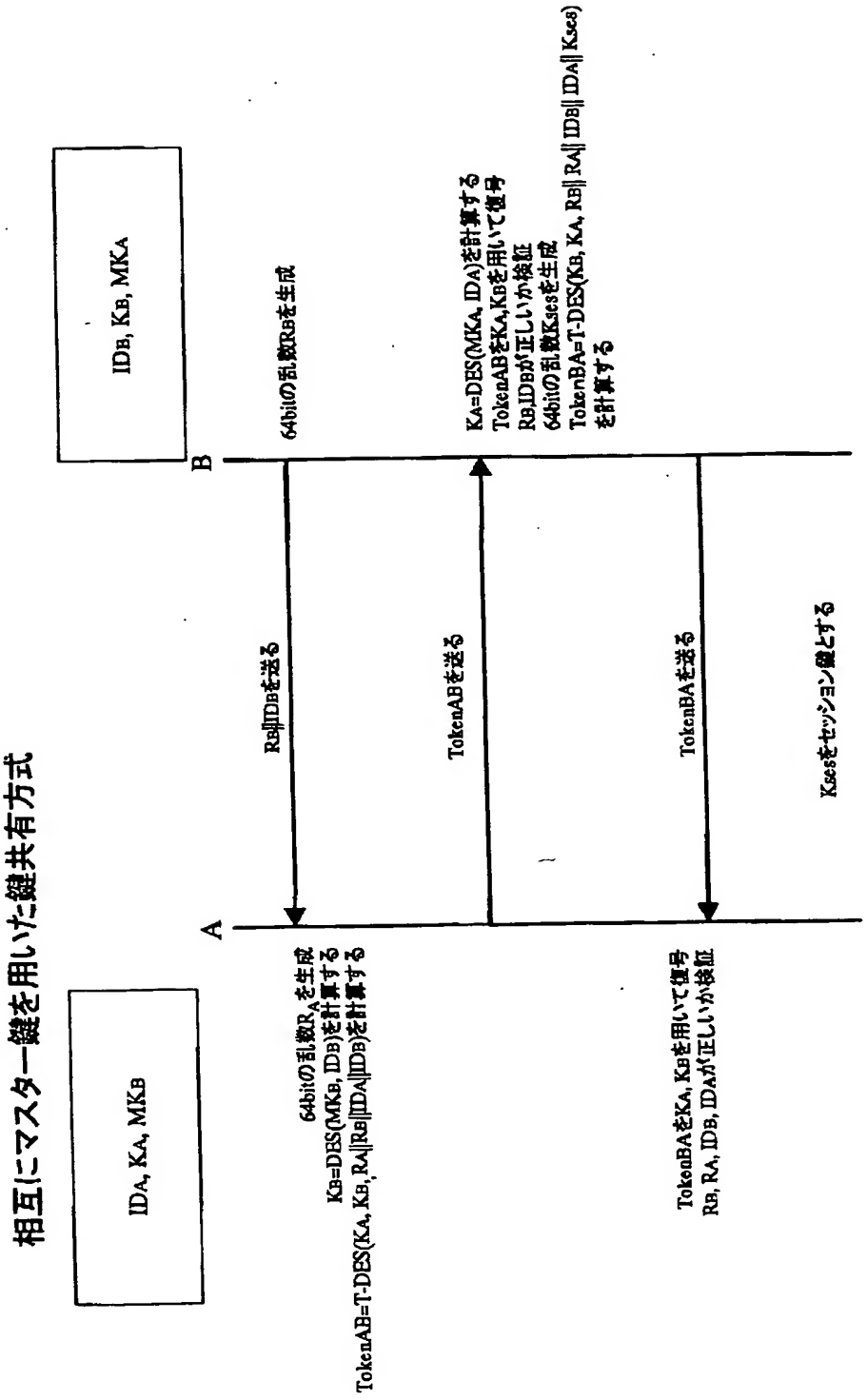


Device認証

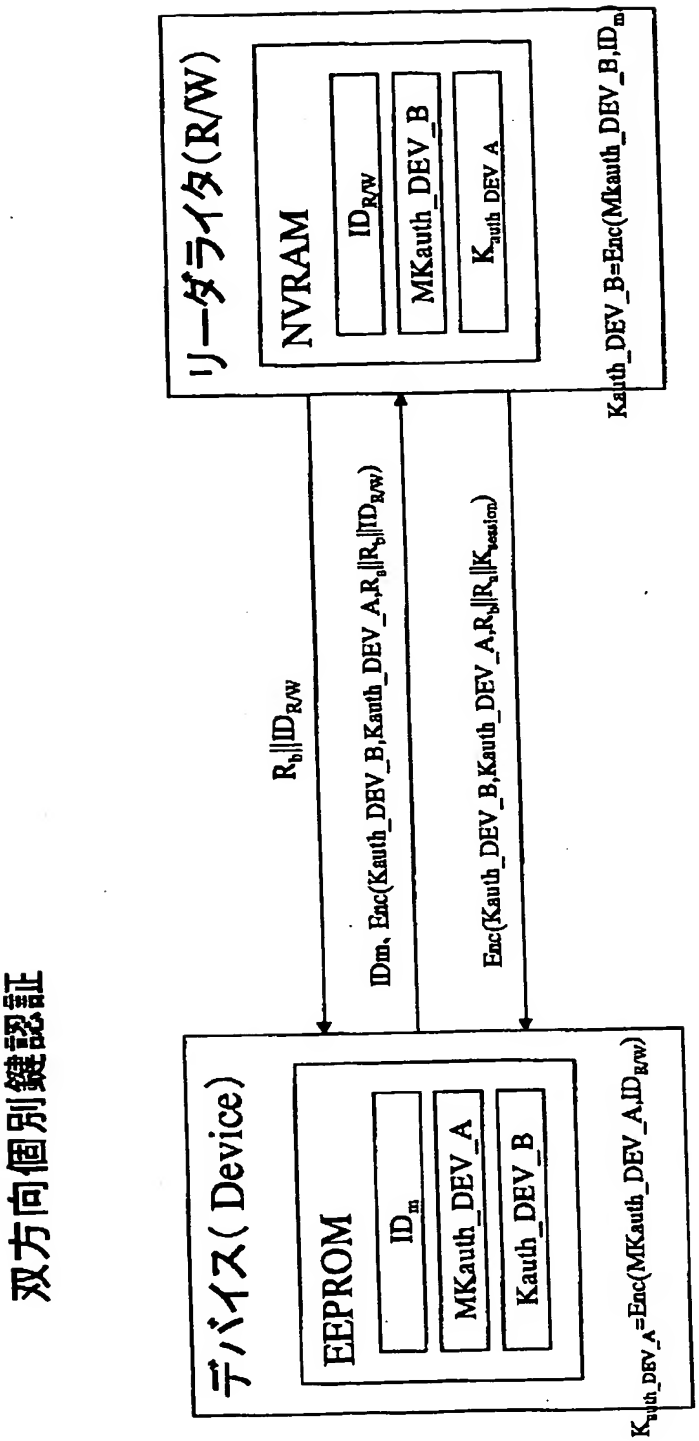
【図５０】


ISO/IEC 9798-3 非対称鍵暗号技術を用いた相互認証および鍵共有方式
A.Sig.Gen.Key(Aの秘密鍵)
A.Certificate(Aの公開鍵証明書)
p、G
B.Sig.Gen.Key(Bの秘密鍵)
B.Certificate(Bの公開鍵証明書)
p、G
A
B
64bitの乱数RBを生成
RBを送る
64bitの乱数RAを生成
乱数AK(0<AK<p)を生成
AV=AK×Gを計算
A.Sig=Sign(A.Sig.Gen.Key、RA||RB||AV)を計算
A.Certificate||RA||RB||AV||A.Sigを送る
Aから送られてきたRBが生成したものと同じか検証
認証局の公開鍵でAの公開鍵証明書を検証
A.Sig.Ver.Key(Aの公開鍵)を入手
A.SigをA.Sig.Ver.Keyで検証
乱数BK(0<BK<p)を生成
BV=BK×Gを計算
B.Sig=Sign(B.Sig.Gen.Key、RB||RA||BV)を計算
BK×AVを計算し、X座標の64BitをSession Keyとする
B.Certificate||RB||RA||BV||B.Sigを送る
Bから送られてきたRAが生成したものと同じか検証
認証局の公開鍵でBの公開鍵証明書を検証
B.Sig.Ver.Key(Bの公開鍵)を入手
B.SigをB.Sig.Ver.Keyで検証
AK×BVを計算し、X座標の64BitをSession Keyとする

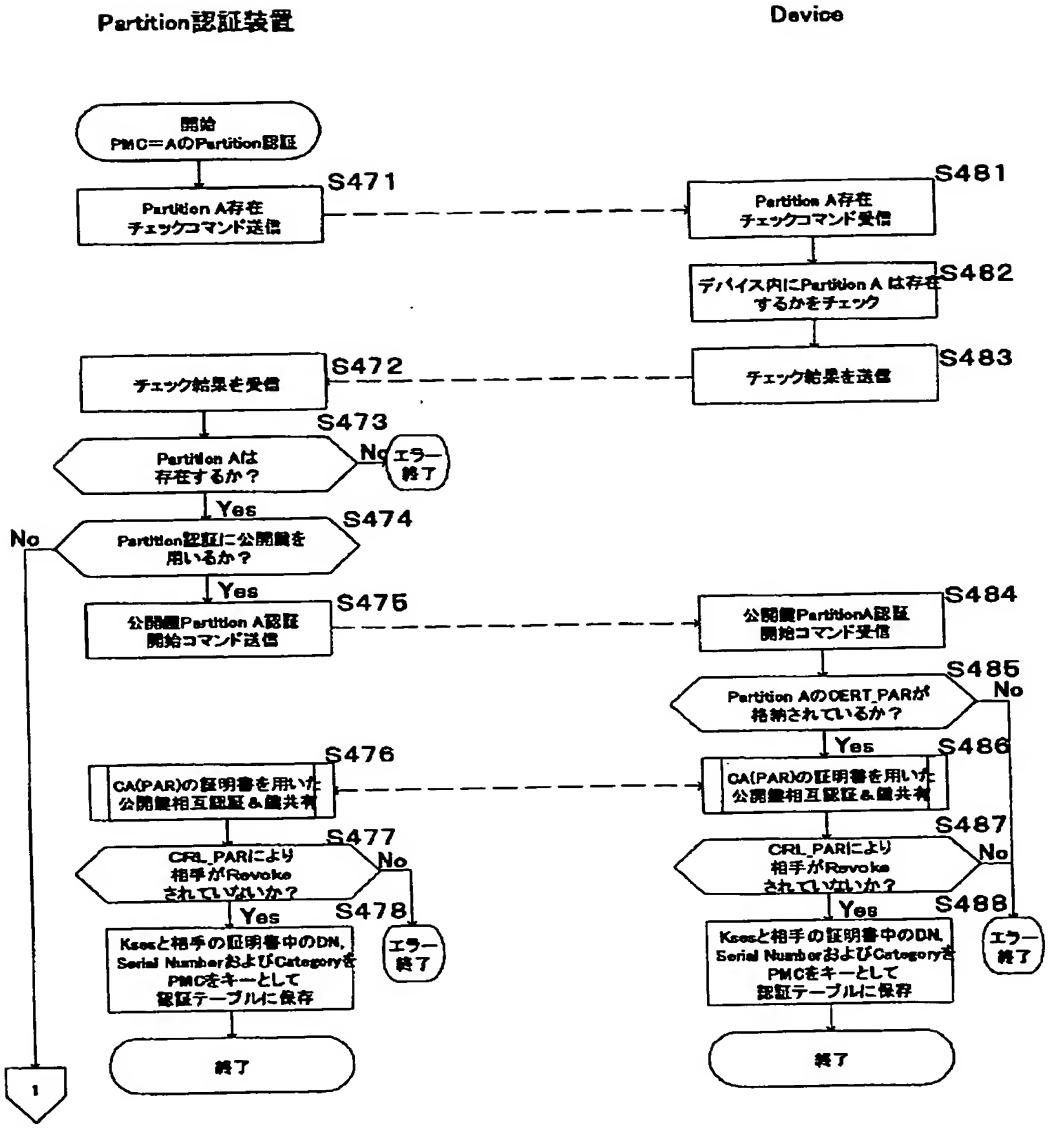
【図５３】


相互にマスター鍵を用いた鍵共有方式
IDA, KA, MKB
IDB, KB, MKA
A
B
64bitの乱数RAを生成
KB=DES(MKB, IDB)を計算する
TokenAB=T-DES(KA, KB, RA‖RB‖IDA‖IDB)を計算する
RB‖IDBを送る
64bitの乱数RBを生成
KA=DES(MKA, IDA)を計算する
TokenABをKA,KBを用いて復号
RB,IDBが正しいか検証
64bitの乱数Ksesを生成
TokenBA=T-DES(KB, KA, RB‖RA‖IDB‖IDA‖Kses)
を計算する
TokenABを送る
TokenBAを送る
TokenBAをKA, KBを用いて復号
RB, RA, IDB, IDAが正しいか検証
Ksesをセッション鍵とする

【図５４】

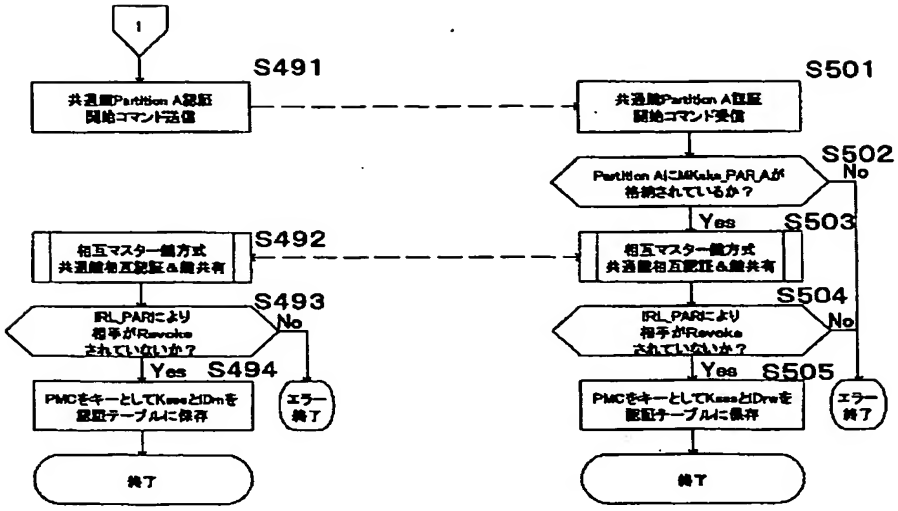

双方向個別鍵認証
デバイス(Device)
EEPROM
ID_m
MKauth_DEV_A
Kauth_DEV_B
$K_{auth\_DEV\_A}$=Enc(MKauth_DEV_A,$ID_{R/W}$)
$R_b||ID_{R/W}$
IDm、Enc(Kauth_DEV_B,Kauth_DEV_A,$R_a||R_b||ID_{R/W}$)
Enc(Kauth_DEV_B,Kauth_DEV_A,$R_a||R_b||K_{session}$)
リーダライタ(R/W)
NVRAM
$ID_{R/W}$
MKauth_DEV_B
$K_{auth\ DEV\ A}$
Kauth_DEV_B=Enc(Mkauth_DEV_B,ID_m)

【図５５】



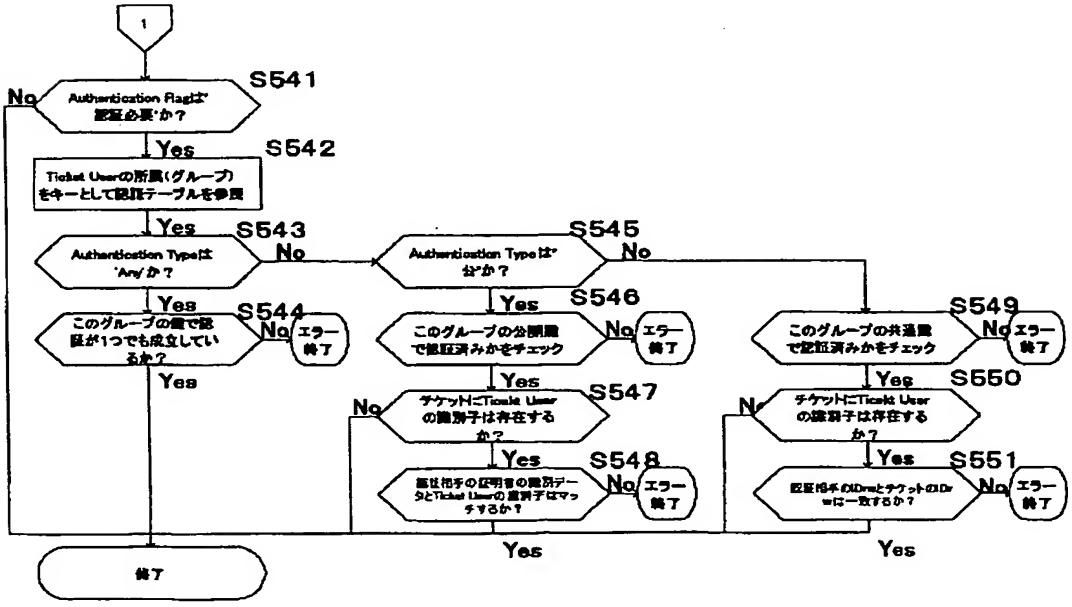
Partition認証

【図５６】



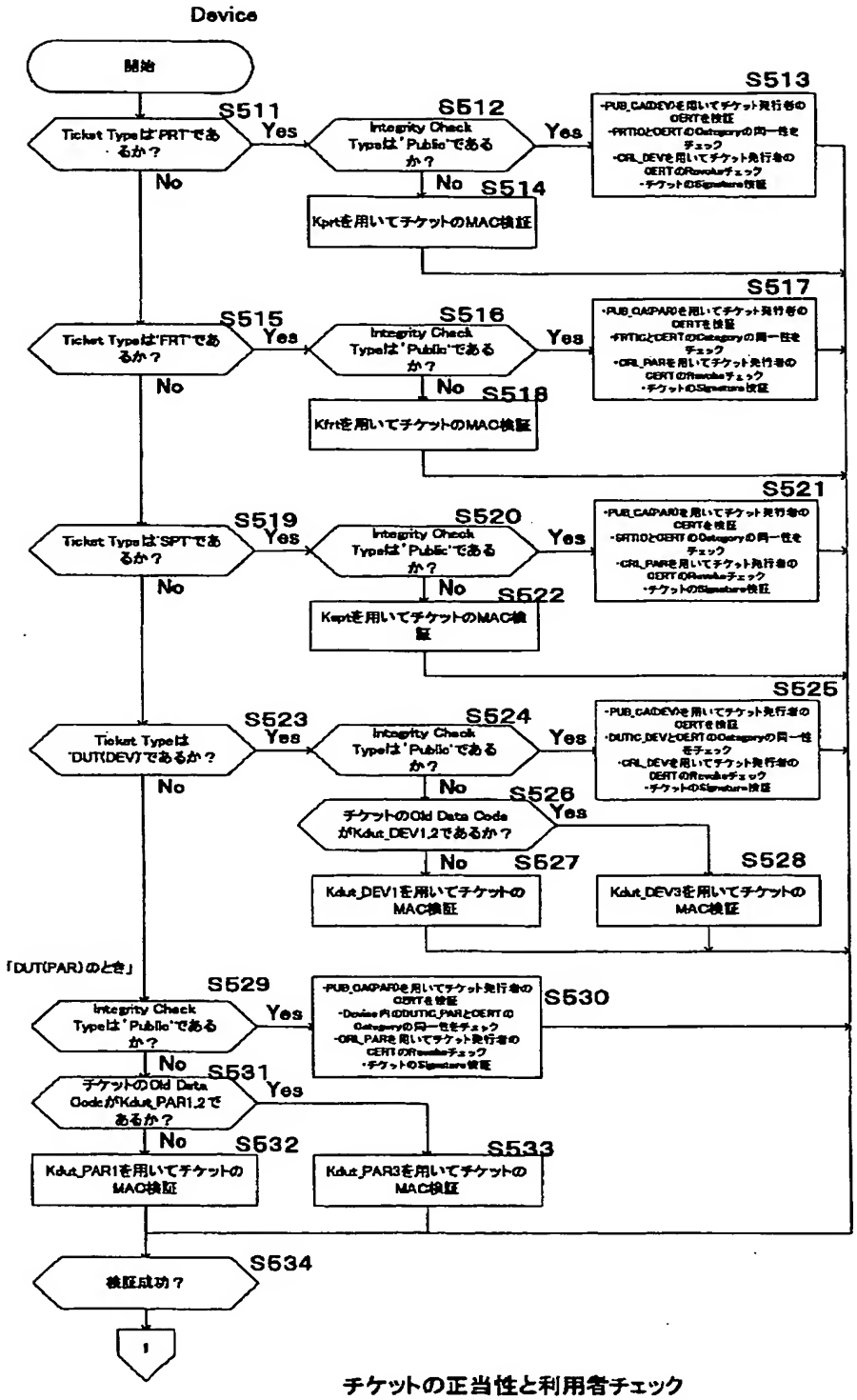
Partition認証（続き）

【図５８】



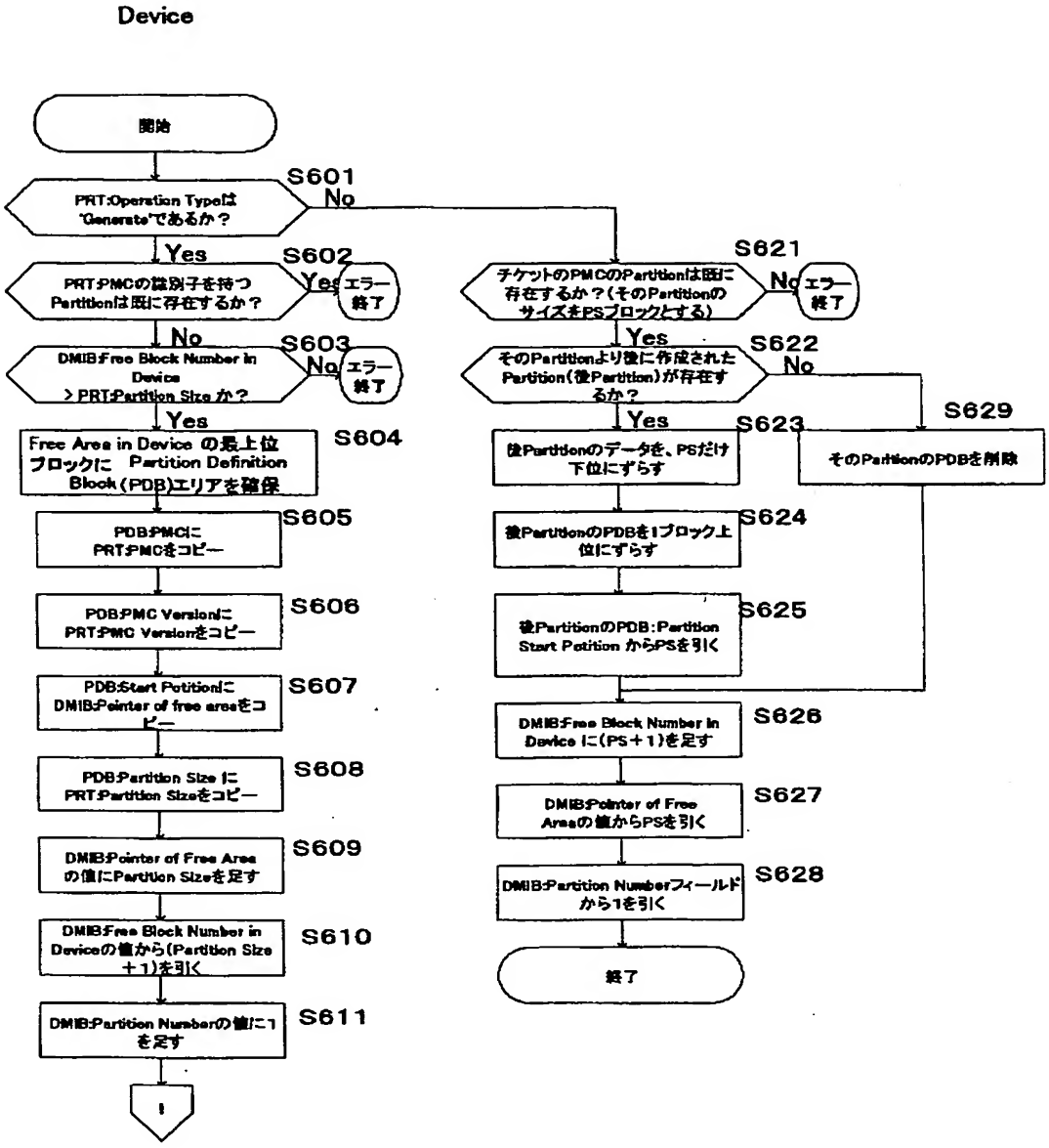
チケットの正当性と利用者チェック（続き）

【図５７】



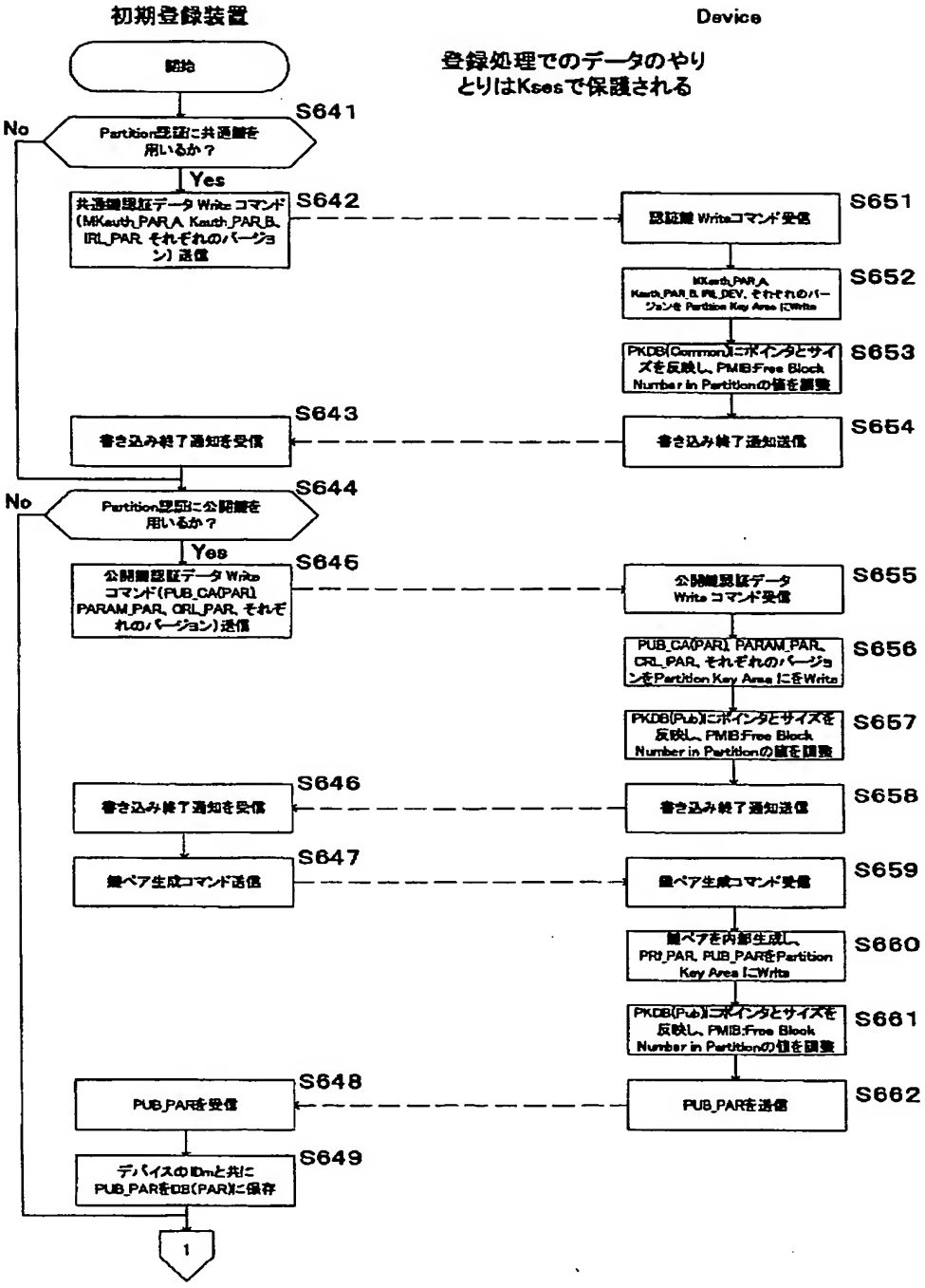
チケットの正当性と利用者チェック

【図６０】



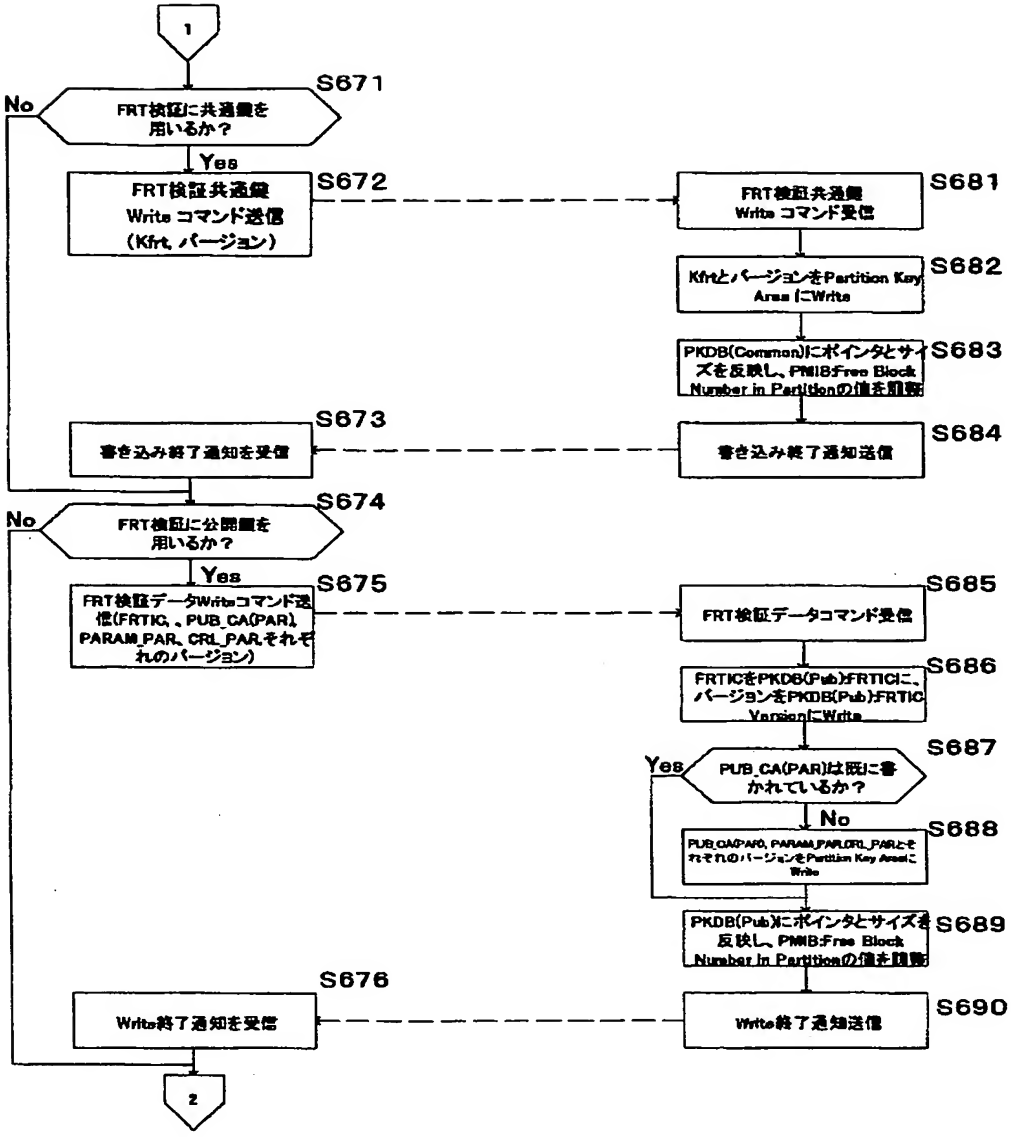
Partition作成・削除操作

【図６２】



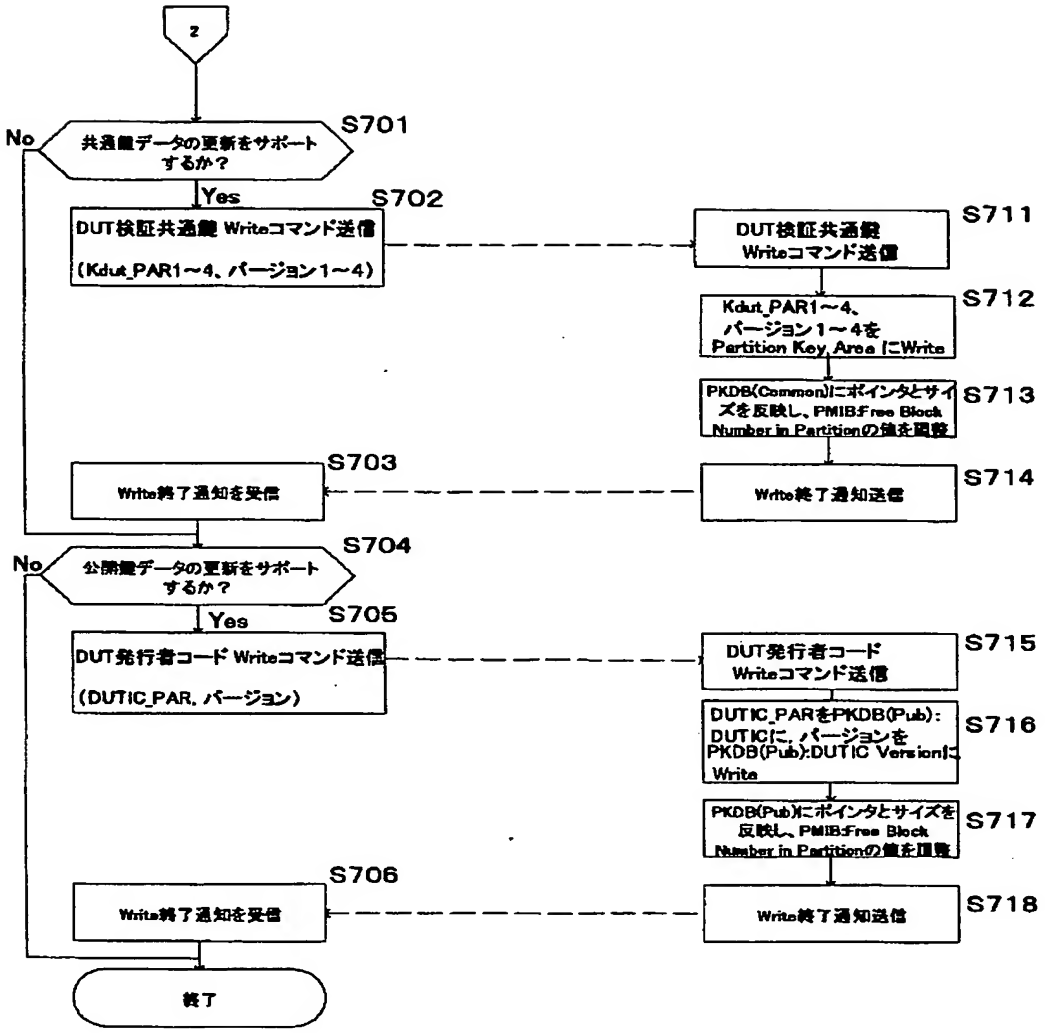
Partition初期登録

【図６３】



Partition初期登録(続き)

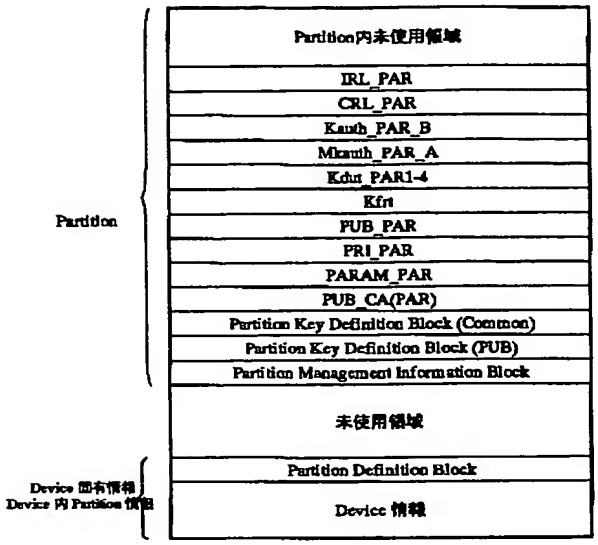
【図６４】



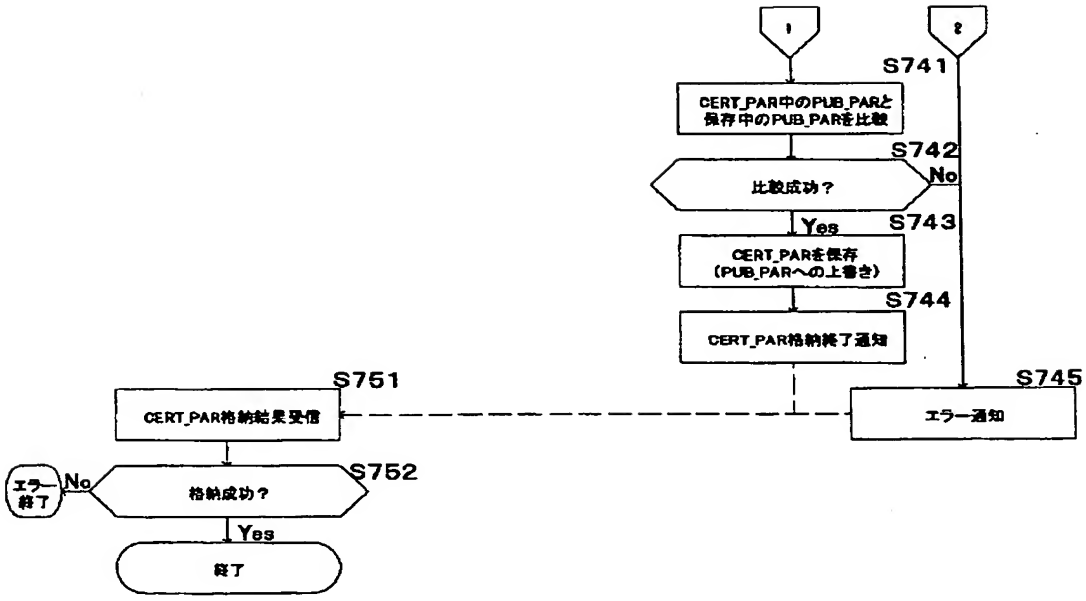
Partition初期登録(続き)

【図６５】

Partition生成後の状態

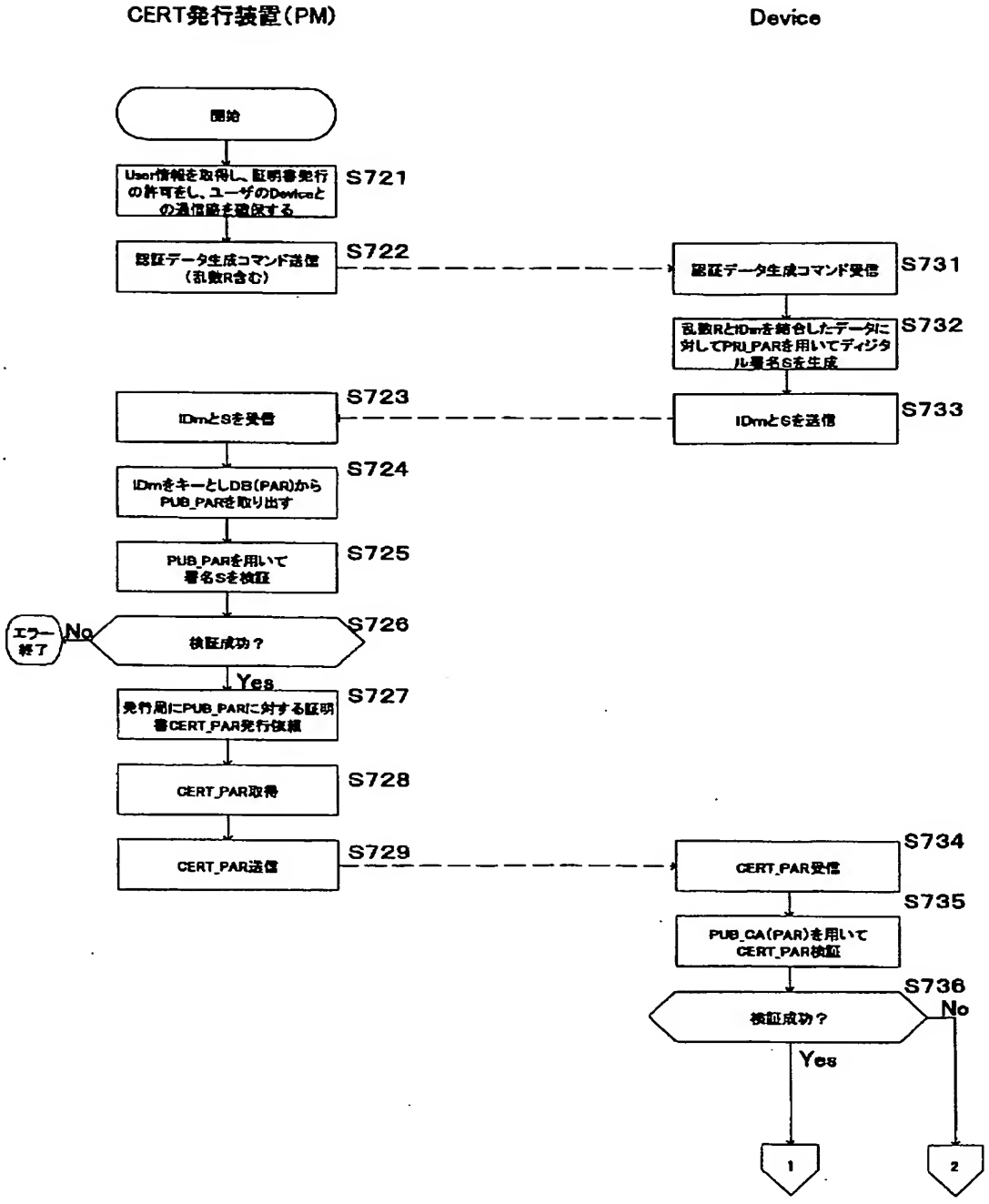


【図６７】



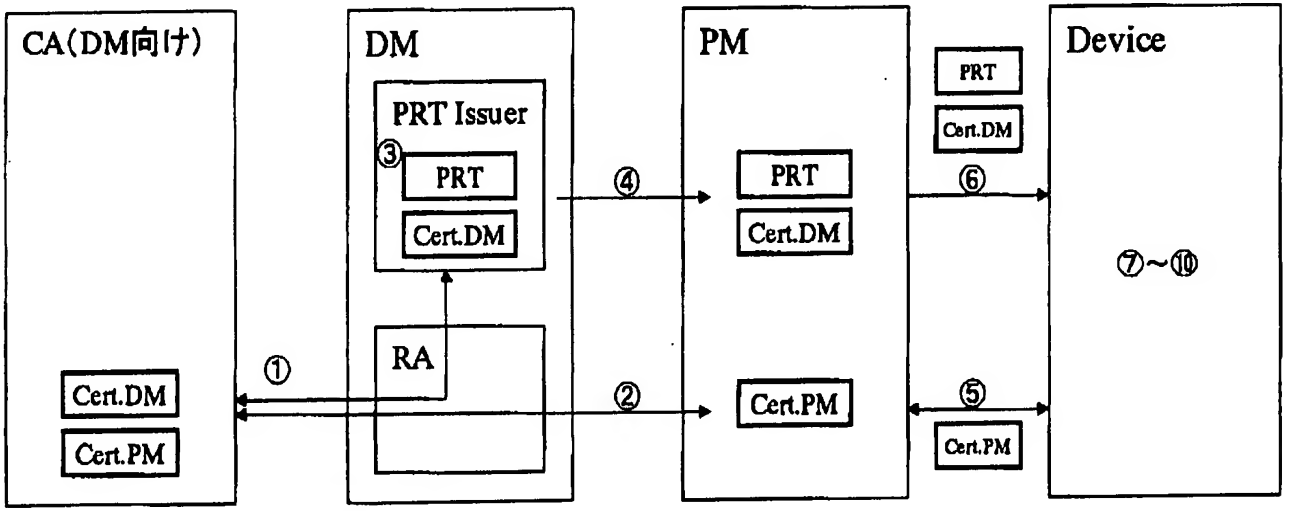
CERT発行(PM)(続き)

【図６６】



CERT発行(PM)

Partition生成手順（Authentication Type: 公開鍵, PRT:公開鍵）



【図６８】

①DM(Device Manager)用の公開鍵証明書の発行
②PM(Partition Manager)用の公開鍵証明書の発行
③PRT(Partition Registration Ticket)の生成
④PRT及びDMのCertificateの供給
　→PRTには、検証値(公開鍵)が付いている
⑤PMとDeviceの間の相互認証(公開鍵)
⑥PRT及びDMのCertificateの送信
　→PRTの検証
　→PRT生成者の検証、PRT使用者の検証
⑦Partitionの生成
⑧鍵データ書き込み
⑨Public Keyの読み出し(作成したPartitionでは公開鍵認証を用いる場合)
⑩Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

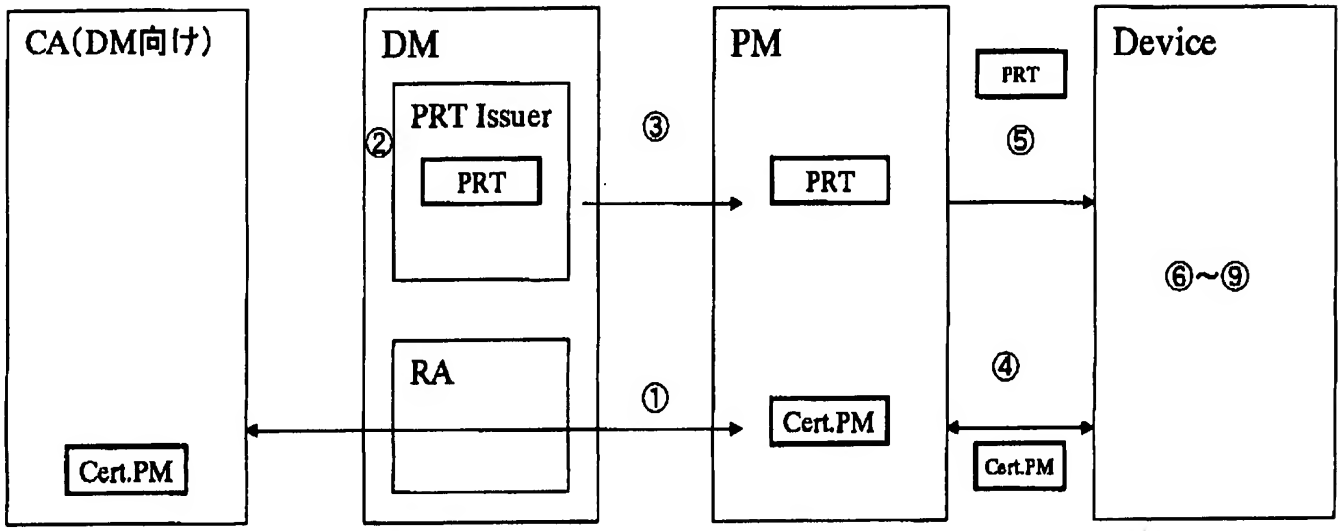
PRT例

PRT Version Number
PRT Size
Authentication Type=公
Category:PRT User
Partition Manager Code
Partition Size
Signature

Certificate例

Certificate Version
SN
DN:Device Manager
・・・
DM Public Key
Group:DMC Category:PRT Issuer
Signature

Partition生成手順（Authentication Type: 公開鍵, PRT:共通鍵）



【図６９】

①PM(Partition Manager)用の公開鍵証明書の発行
②PRT(Partition Registration Ticket)の生成
③PRTの供給
　→PRTには、検証値(共通鍵)が付いている
④PMとDeviceの間の相互認証(公開鍵)
⑤PRTの送信
　→PRTの検証
　→PRT生成者の検証、PRT使用者の検証
⑥Partitionの生成
⑦鍵データ書き込み
⑧Public Keyの読み出し
(作成したPartitionでは公開鍵認証を用いる場合)
⑨Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

PRT例

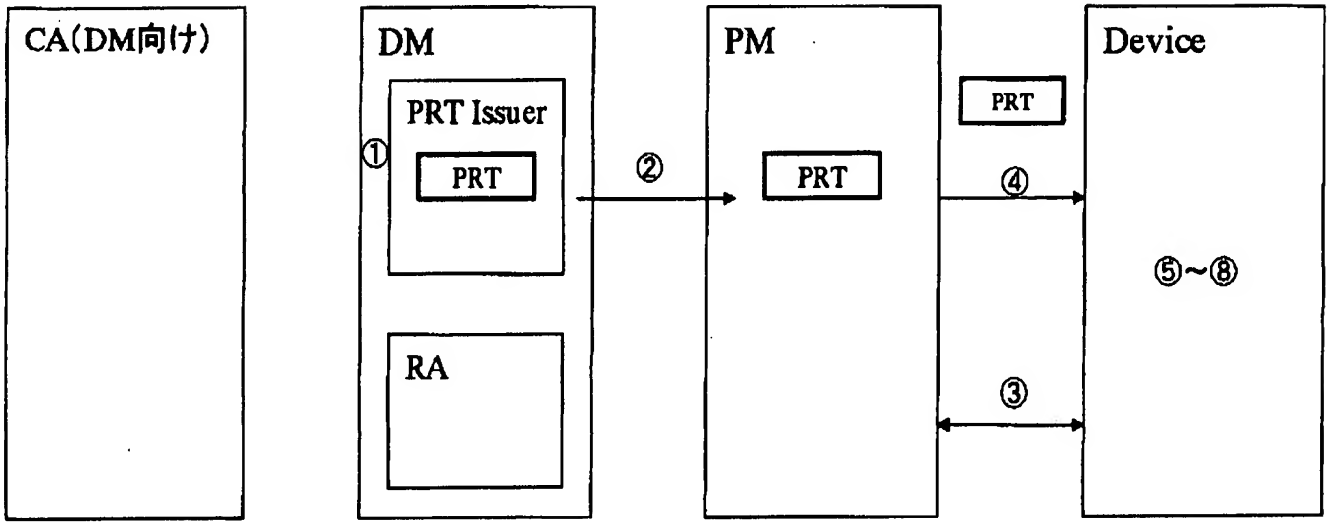
PRT Version Number
PRT Size
Authentication Type=公
Category:PRT User
Partition Manager Code
Partition Size
MAC

Certificate例

Certificate Version
SN
DN:Device Manager
・・・
DM Public Key
Group:DMC Category:PRT Issuer
Signature

特開平14-278838

Partition生成手順（Authentication Type: 共通鍵, PRT:共通鍵）



【図７０】

①PRT(Partition Registration Ticket)の生成
②PRTの供給
　→PRTには、検証値(共通鍵)が付いている
③PMとDeviceの間の相互認証(共通鍵)
④PRTの送信
　→PRTの検証
　→PRT生成者の検証、PRT使用者の検証
⑤Partitionの生成
⑥鍵データ書き込み
⑦Public Keyの読み出し
（作成したPartitionでは公開鍵認証を用いる場合）
⑧Certificateの発行（作成したPartitionでは公開鍵認証を用いる場合）

PRT例

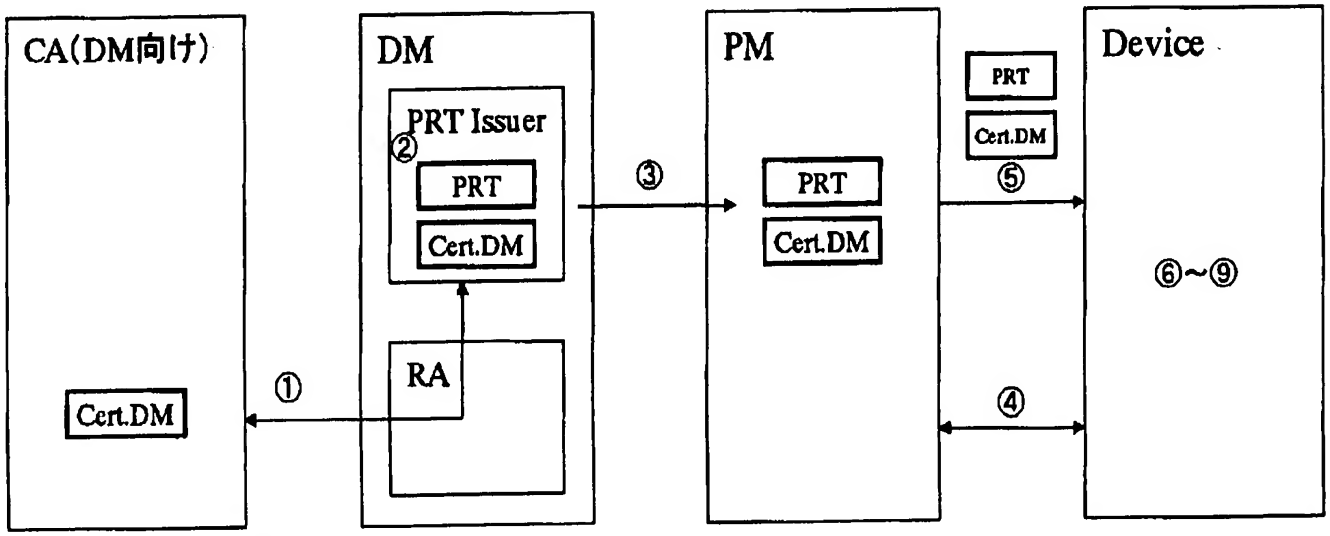
PRT Version Number
PRT Size
Authentication Type=共
Category:PRT User
Partition Manager Code
Partition Size
MAC

Certificate例

Certificate Version
SN
DN:Device Manager
・・・
DM Public Key
Group:DMC Category:PRT Issuer
Signature

特開平14-278838

Partition生成手順（Authentication Type: 共通鍵, PRT:公開鍵）



【図７１】

①DM(Device Manager)用の公開鍵証明書の発行
②PRT(Partition Registration Ticket)の生成
③PRT及びDMのCertificateの供給
　→PRTには、検証値(公開鍵)が付いている
④PMとDeviceの間の相互認証(共通鍵)
⑤PRT及びDMのCertificateの送信
　→PRTの検証
　→PRT生成者の検証、PRT使用者の検証
⑥Partitionの生成
⑦鍵データ書き込み
⑧Public Keyの読み出し
(作成したPartitionでは公開鍵認証を用いる場合)
⑨Certificateの発行(作成したPartitionでは公開鍵認証を用いる場合)

PRT例

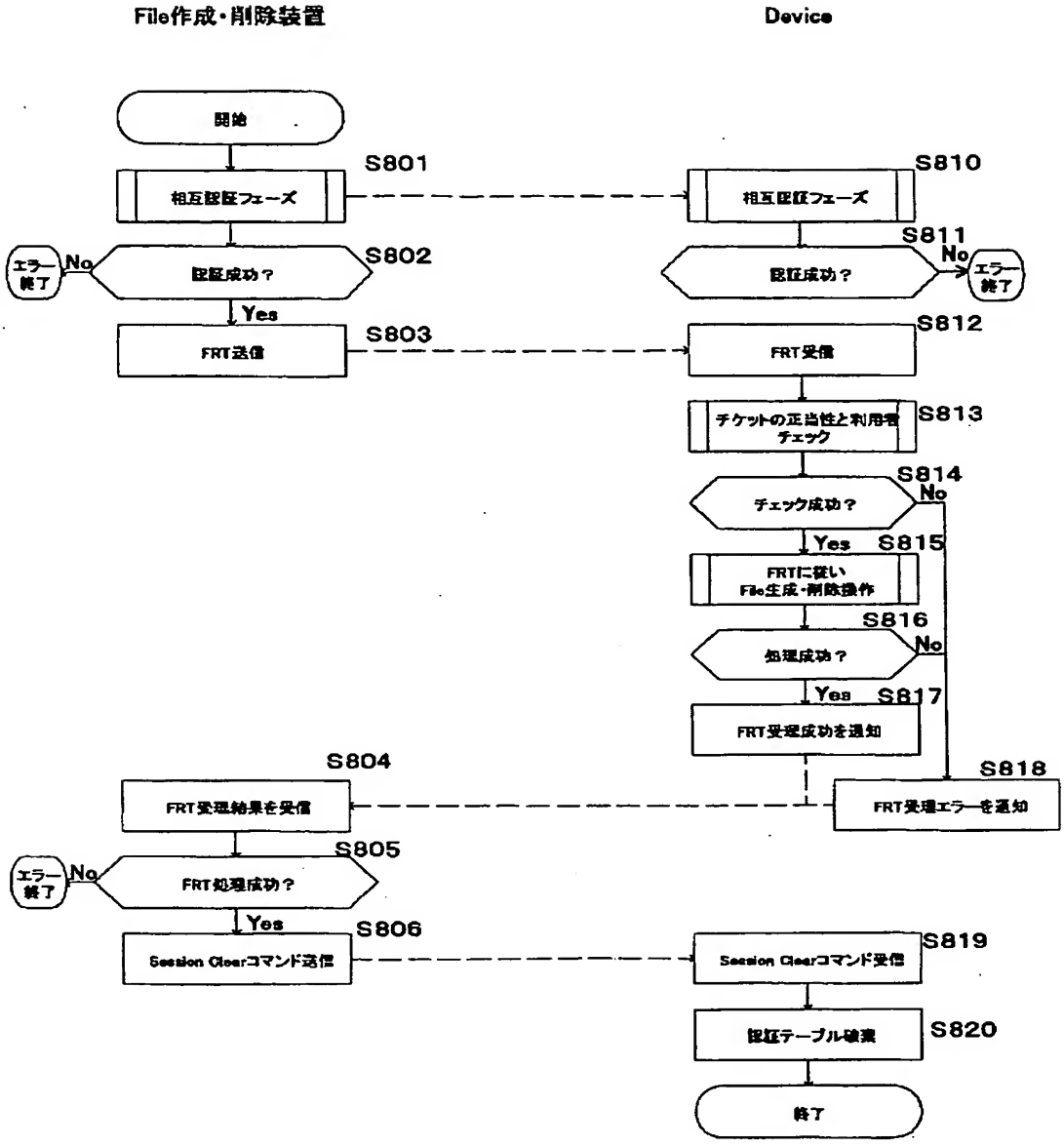
PRT Version Number
PRT Size
Authentication Type=共
Category:PRT User
Partition Manager Code
Partition Size
Signature

Certificate例

Certificate Version
SN
DN:Device Manager
・・・
DM Public Key
Group:DMC Category:PRT Issuer
Signature

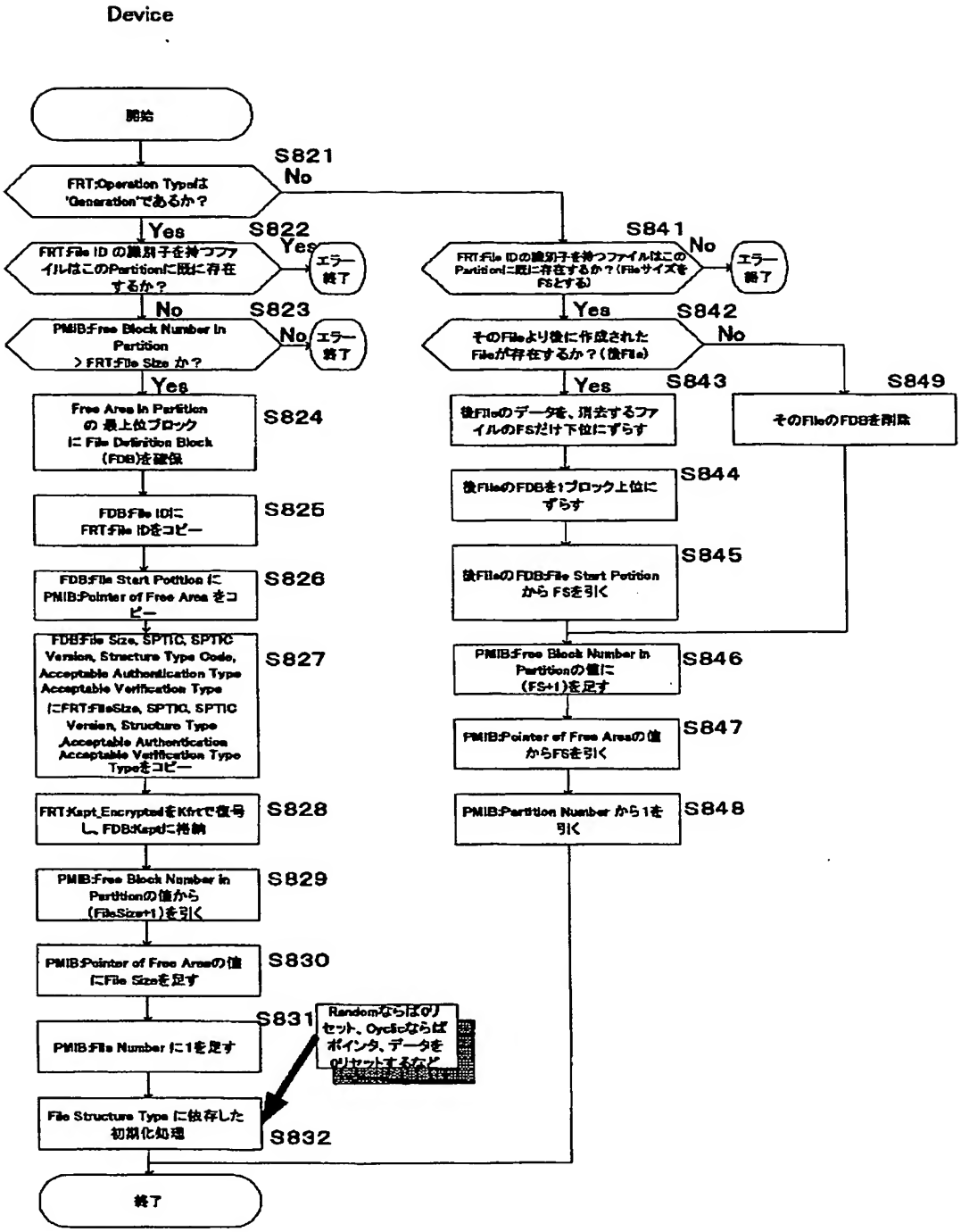
特開平14-278838

【図７２】



File生成・消去

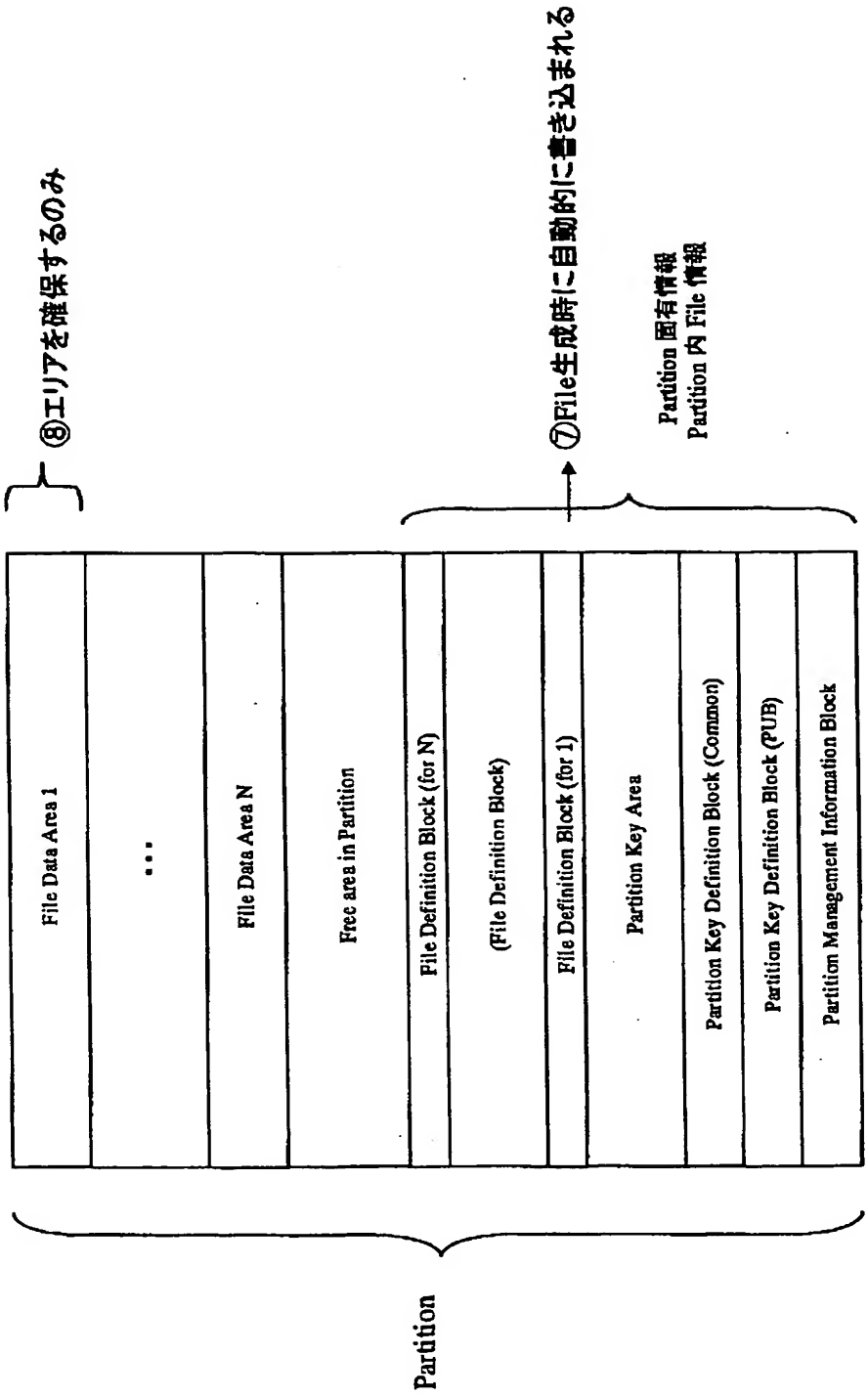
【図７３】



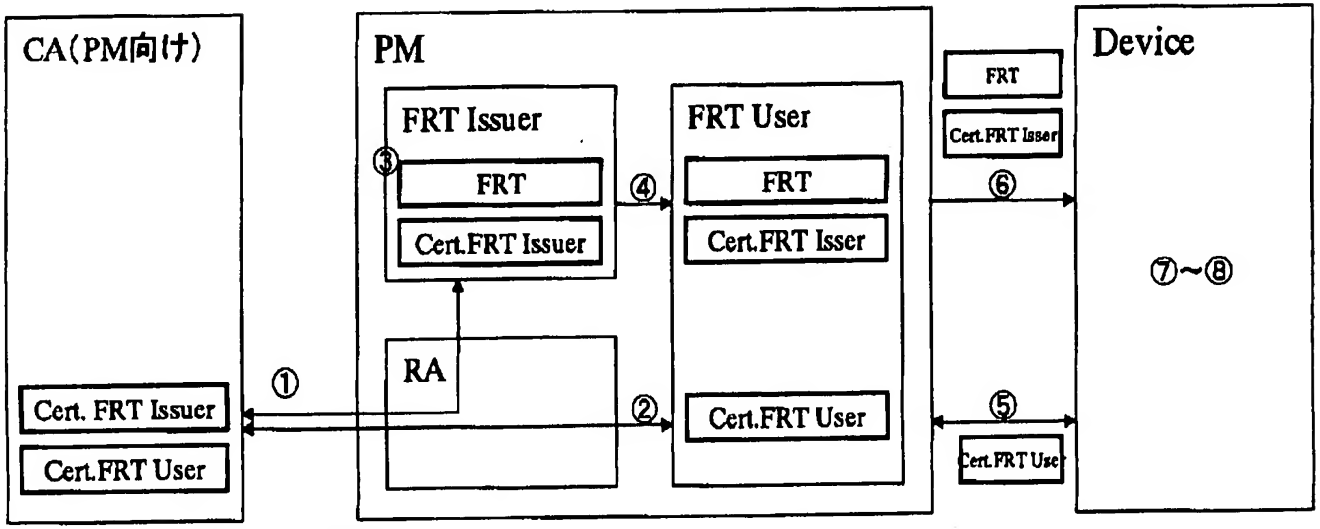
File作成・削除操作

【図７４】

File生成後の状態



File生成手順（Authentication Type: 公開鍵, FRT:公開鍵）



①FRT Issuer用の公開鍵証明書の発行
②FRT User用の公開鍵証明書の発行
③FRT(File Registration Ticket)の生成
④FRT及びFRT Issuer用のCertificateの供給
　→FRTには、検証値(公開鍵)が付いている
⑤FRT IssuerとDeviceの間の相互認証(公開鍵)
⑥FRT及びFRT Issuer用のCertificateの送信
　→FRTの検証
　→FRT生成者の検証、FRT使用者の検証
⑦File Definition BlockにSPT Issuer CategoryとKsptを登録
⑧File Data Areaを確保

FRT例

...
Authentication Type=公
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kfrt(Kspt)
Signature

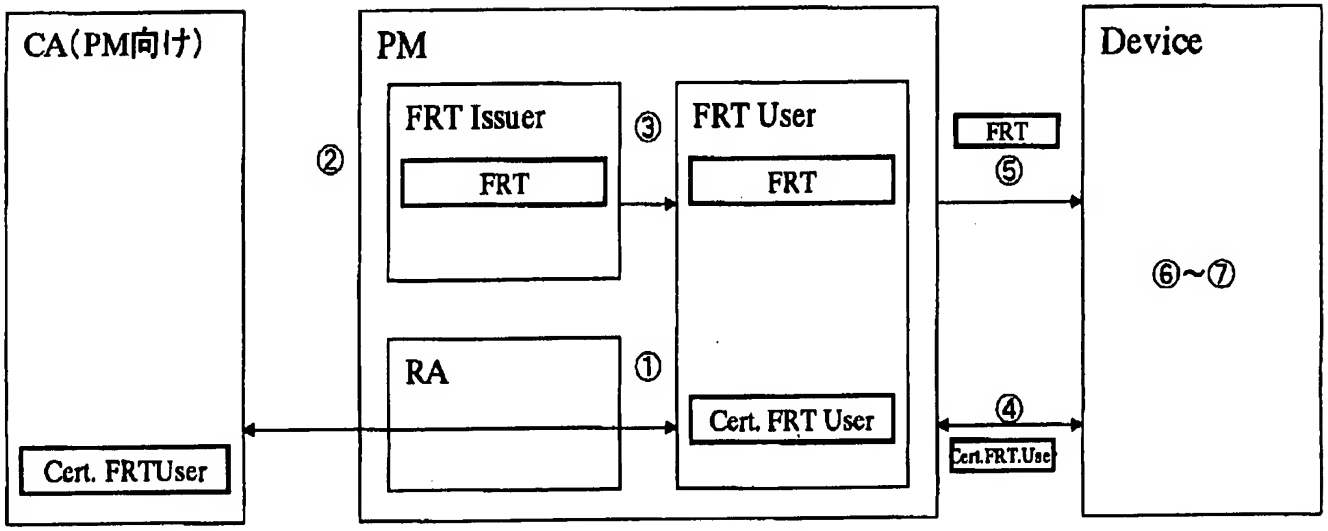
Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC Category:FRT Issuer
Signature

【図７５】

特開平14-278838

File生成手順（Authentication Type: 公開鍵, FRT:共通鍵）



【図７６】

①FRT User用の公開鍵証明書の発行
②FRT(File Registration Ticket)の生成
③FRTの供給
　→FRTには、検証値(共通鍵)が付いている
④FRT IssuerとDeviceの間の相互認証(公開鍵)
⑤FRTの送信
　→FRTの検証
　→FRT生成者の検証、FRT使用者の検証
⑥File Definition BlockにSPT Issuer CategoryとKsptを登録
⑦File Data Areaを確保

FRT例

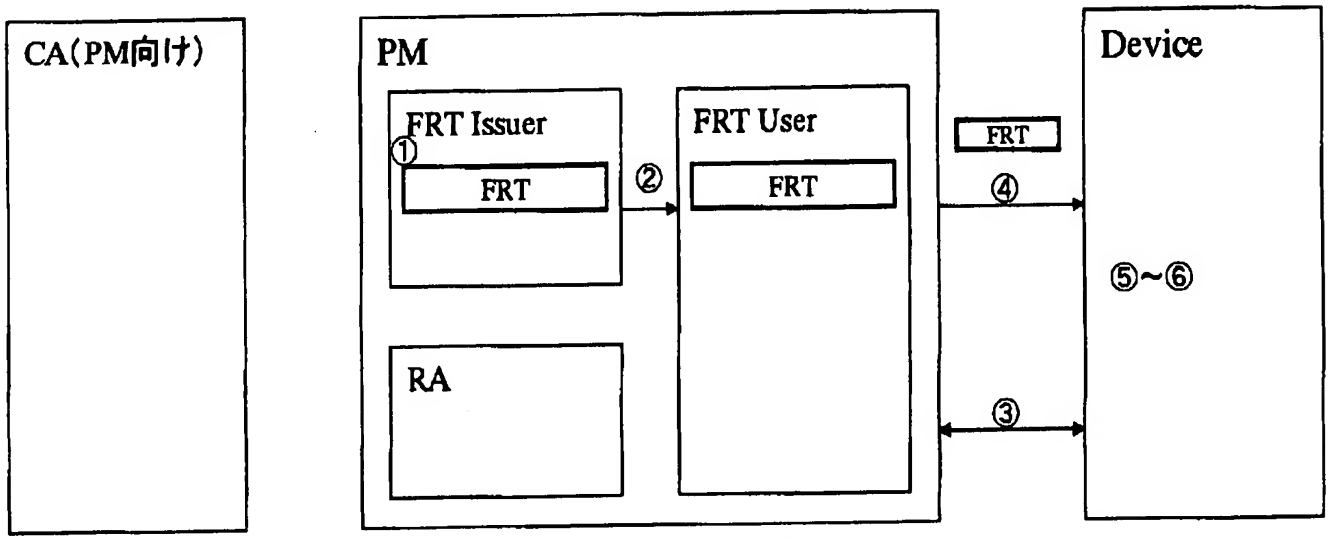
...
Authentication Type=公
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kfrt(Kspt)
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC Category:FRT Issuer
Signature

【図７７】

File生成手順（Authentication Type: 共通鍵, FRT:共通鍵）



①FRT(File Registration Ticket)の生成
②FRTの供給
　→FRTには、検証値(共通鍵)が付いている
③FRT IssuerとDeviceの間の相互認証(共通鍵)
④FRTの送信
　→FRTの検証
　→FRT生成者の検証、FRT使用者の検証
⑤File Definition BlockにSPT Issuer CategoryとKsptを登録
⑥File Data Areaを確保

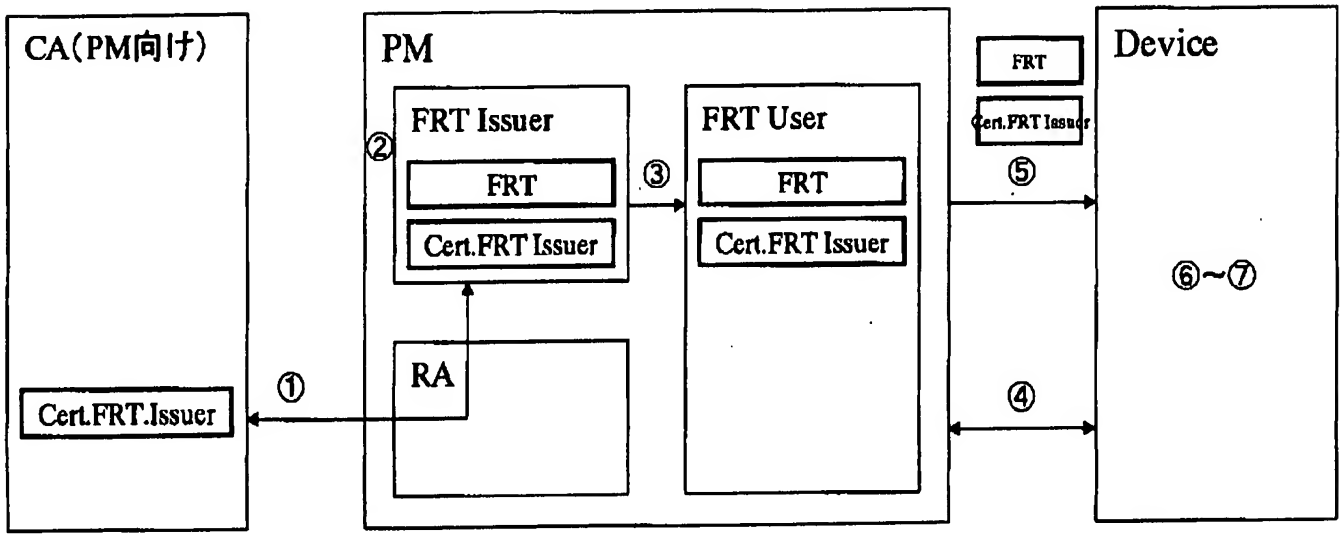
FRT例

...
Authentication Type=共
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kfrt(Kspt)
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC Category:FRT Issuer
Signature

File生成手順（Authentication Type: 共通鍵, FRT:公開鍵）



【図７８】

①FRT Issuer用の公開鍵証明書の発行
②FRT(File Registration Ticket)の生成
③FRT及びFRT Issuer用のCertificateの供給
　→FRTには、検証値(公開鍵)が付いている
④FRT IssuerとDeviceの間の相互認証(共通鍵)
⑤FRT及びFRT Issuer用のCertificateの送信
　→FRTの検証
　→FRT生成者の検証、FRT使用者の検証
⑥File Definition BlockにSPT Issuer CategoryとKsptを登録
⑦File Data Areaを確保

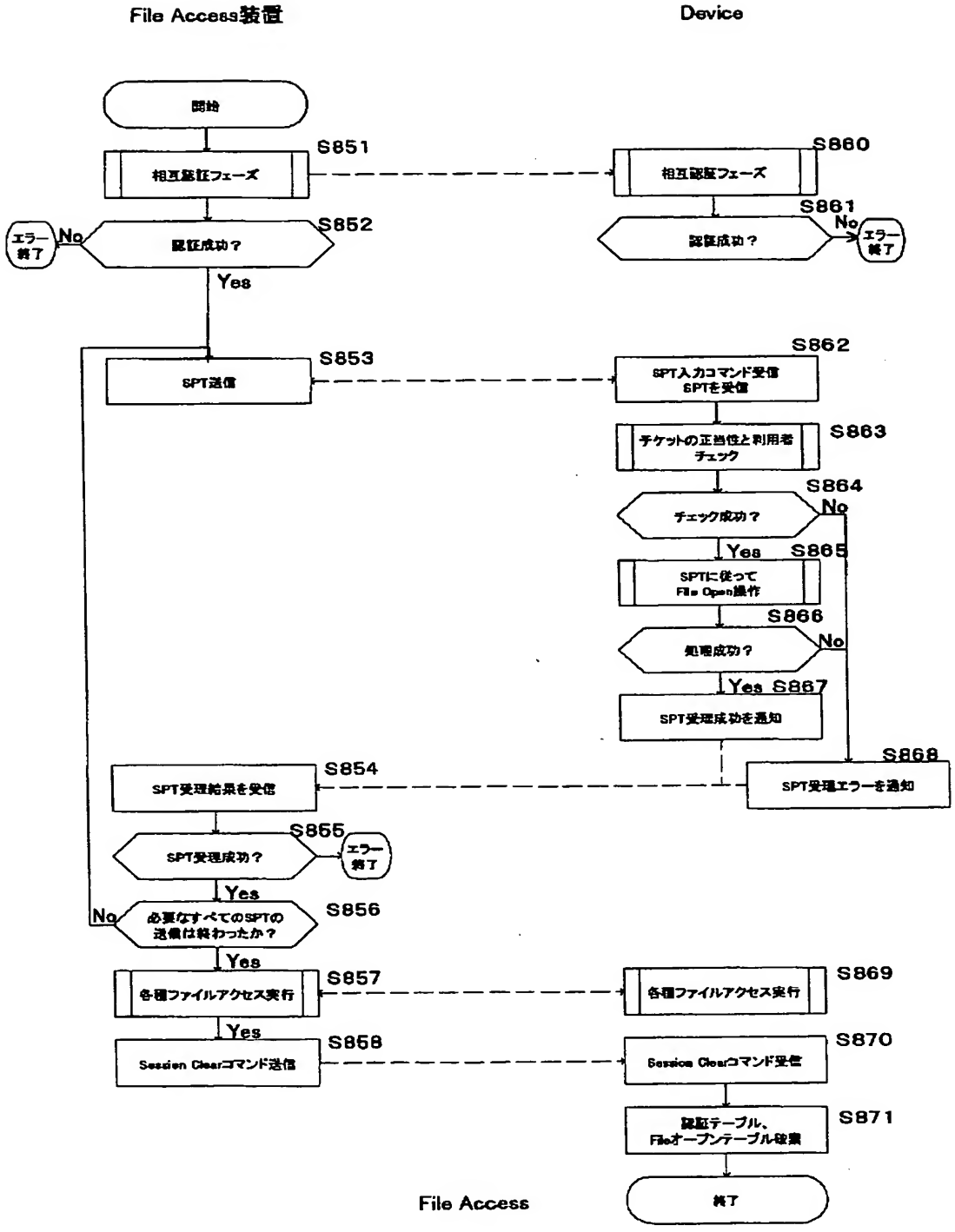
FRT例

...
Authentication Type=共
Category:FRT User
Partition Manager Code
File Information
Category:SPT Issuer
Kfrt(Kspt)
Signature

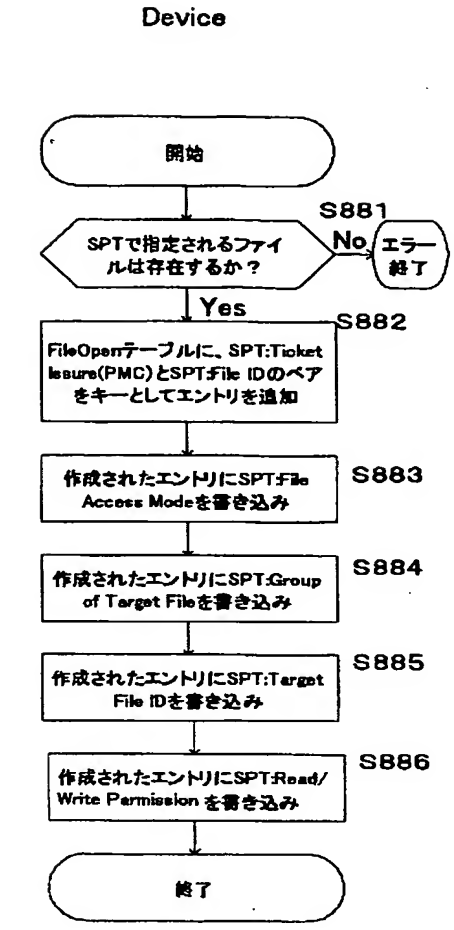
Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM Public Key
Group:PMC Category:FRT Issuer
Signature

【図７９】


File Access装置
Device
開始
S851
相互認証フェーズ
S860
相互認証フェーズ
S852
認証成功？
No
エラー終了
S861
認証成功？
No
エラー終了
Yes
S853
SPT送信
S862
SPT入力コマンド受信
SPTを受信
S863
チケットの正当性と利用者チェック
S864
チェック成功？
No
Yes
S865
SPTに従ってFile Open操作
S866
処理成功？
No
Yes
S867
SPT受理成功を通知
S868
SPT受理エラーを通知
S854
SPT受理結果を受信
S855
SPT受理成功？
エラー終了
Yes
No
必要なすべてのSPTの送信は終わったか？
S856
Yes
S857
各種ファイルアクセス実行
S869
各種ファイルアクセス実行
Yes
S858
Session Clearコマンド送信
S870
Session Clearコマンド受信
S871
認証テーブル、Fileオープンテーブル破棄
終了
File Access

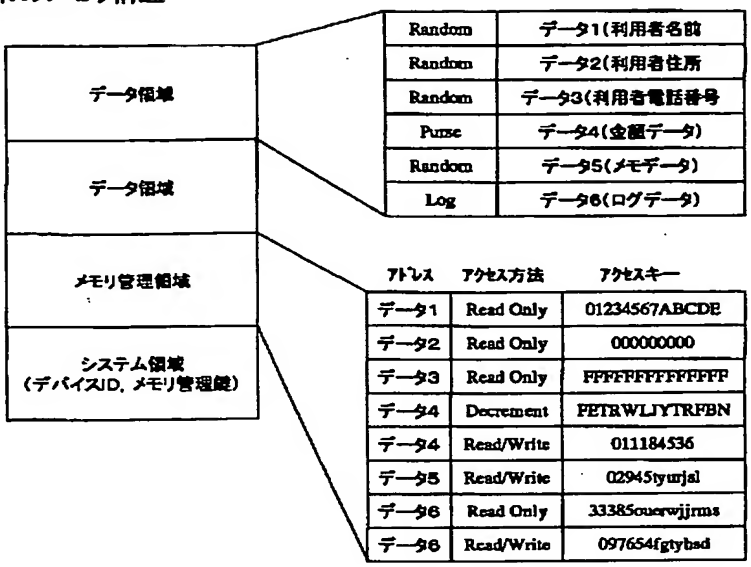
【図８０】



SPTに従ってFile Open操作

【図９６】

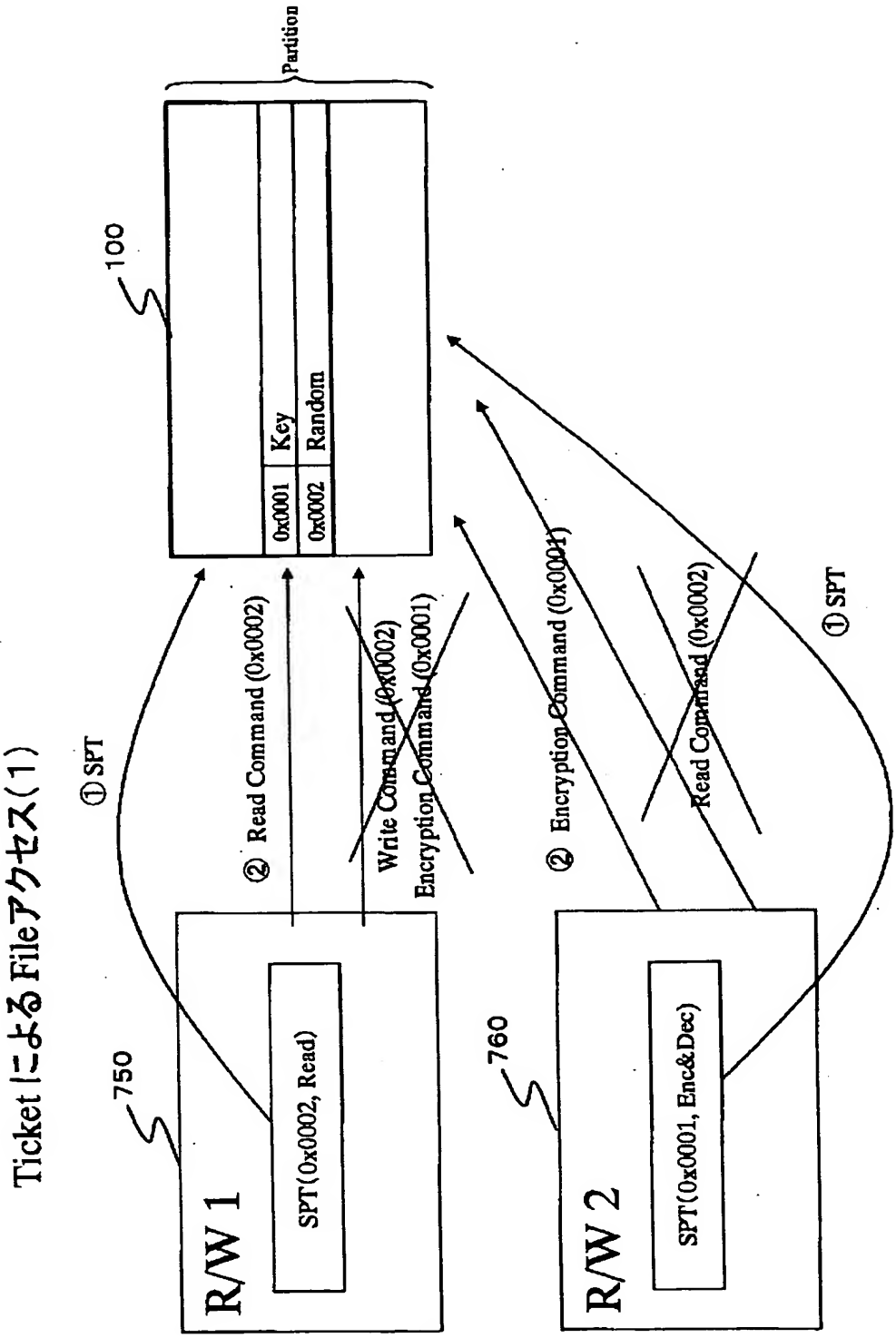
従来のメモリ構造



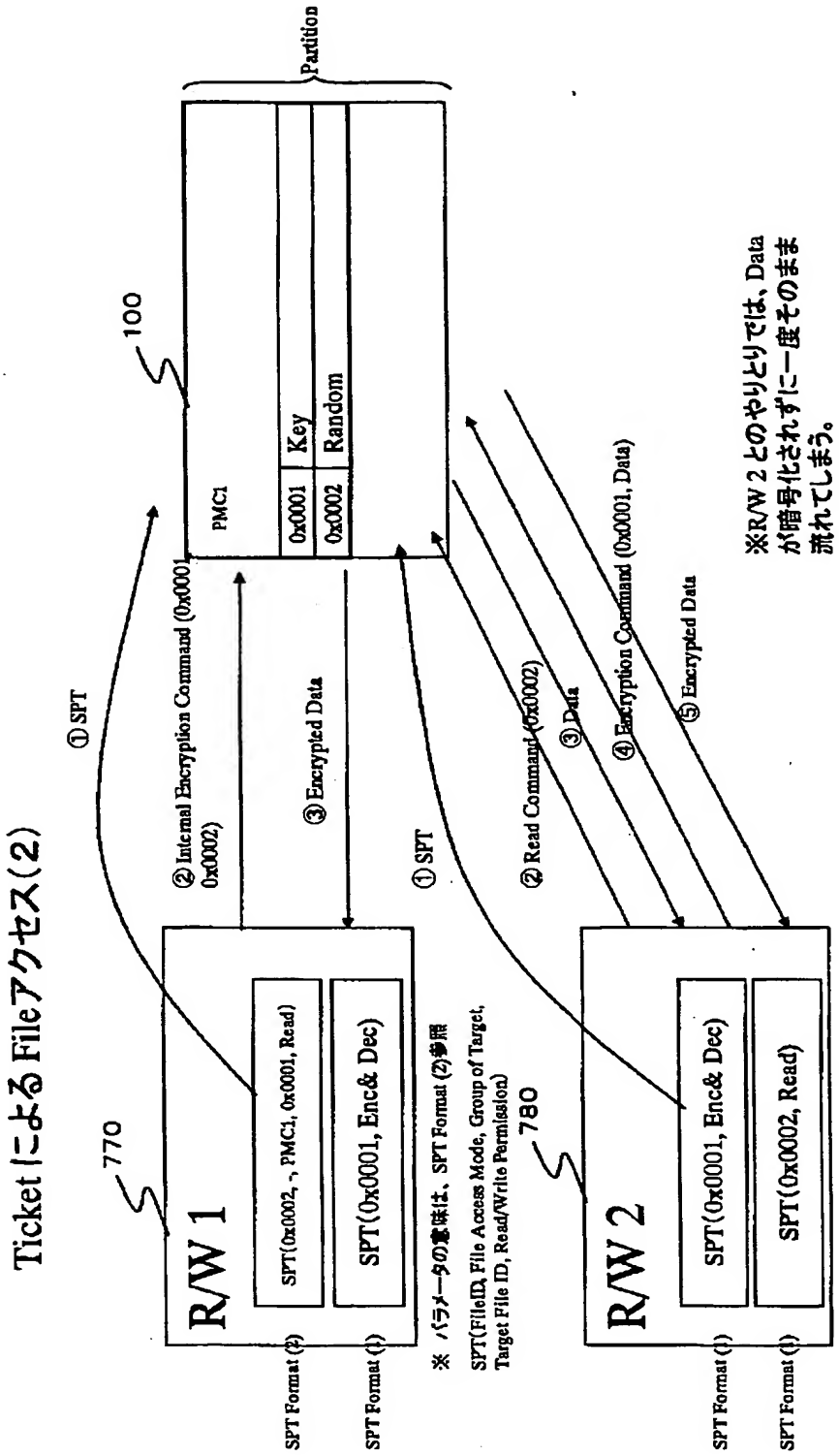
Random	データ1(利用者名前	
Random	データ2(利用者住所	
Random	データ3(利用者電話番号	
Purse	データ4(金額データ)	
Random	データ5(メモデータ)	
Log	データ6(ログデータ)	

アドレス	アクセス方法	アクセスキー
データ1	Read Only	01234567ABCDE
データ2	Read Only	000000000
データ3	Read Only	FFFFFFFFFFFFF
データ4	Decrement	FETRWLIYTRFBN
データ4	Read/Write	011184536
データ5	Read/Write	02945tyurjal
データ6	Read Only	33385ouerwjjrms
データ6	Read/Write	097654fgtybad

【図８３】

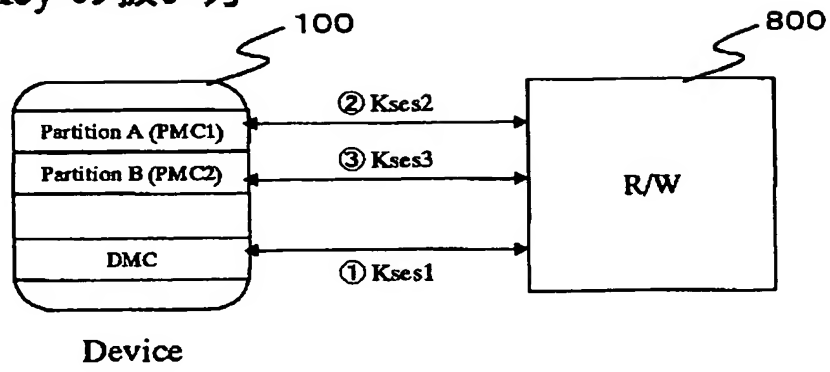

Ticket によるFileアクセス(1)
100
Partition
0x0001
Key
0x0002
Random
① SPT
② Read Command (0x0002)
Write Command (0x0002)
Encryption Command (0x0001)
② Encryption Command (0x0001)
Read Command (0x0002)
① SPT
750
R/W 1
SPT(0x0002, Read)
760
R/W 2
SPT(0x0001, Enc&Dec)

【図８４】


Ticket による File アクセス(2)
100
Partition
PMC1
0x0001
Key
0x0002
Random
① SPT
② Internal Encryption Command (0x0001, 0x0002)
③ Encrypted Data
① SPT
② Read Command (0x0002)
③ Data
④ Encryption Command (0x0001, Data)
⑤ Encrypted Data
※R/W 2 とのやりとりでは、Data が暗号化されずに一度そのまま流れてしまう。
770
R/W 1
SPT(0x0002, -, PMC1, 0x0001, Read)
SPT(0x0001, Enc& Dec)
SPT Format (2)
SPT Format (1)
※ パラメータの意味は、SPT Format (2)参照
SPT(FileID, File Access Mode, Group of Target, Target File ID, Read/Write Permission)
780
R/W 2
SPT(0x0001, Enc& Dec)
SPT(0x0002, Read)
SPT Format (1)
SPT Format (1)

【図８５】

複数 Session Key の扱い方



1. Session Key ＝ Kses1 ⊕ Kses2 ⊕ Kses3

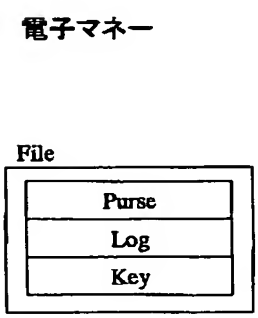
2. Session Key ＝ Kses3 ※一番最後のものを全体のSession Key として使う

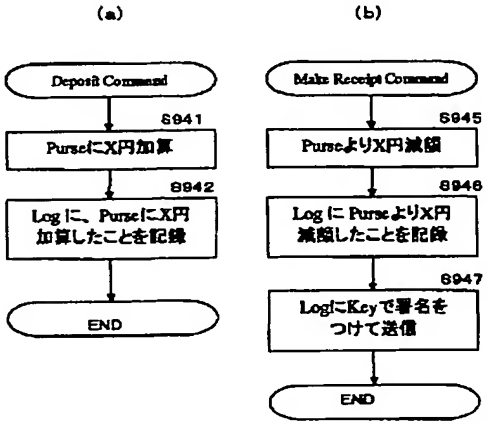
⊕ ： 排他的論理和処理(8バイト単位)

【図８８】

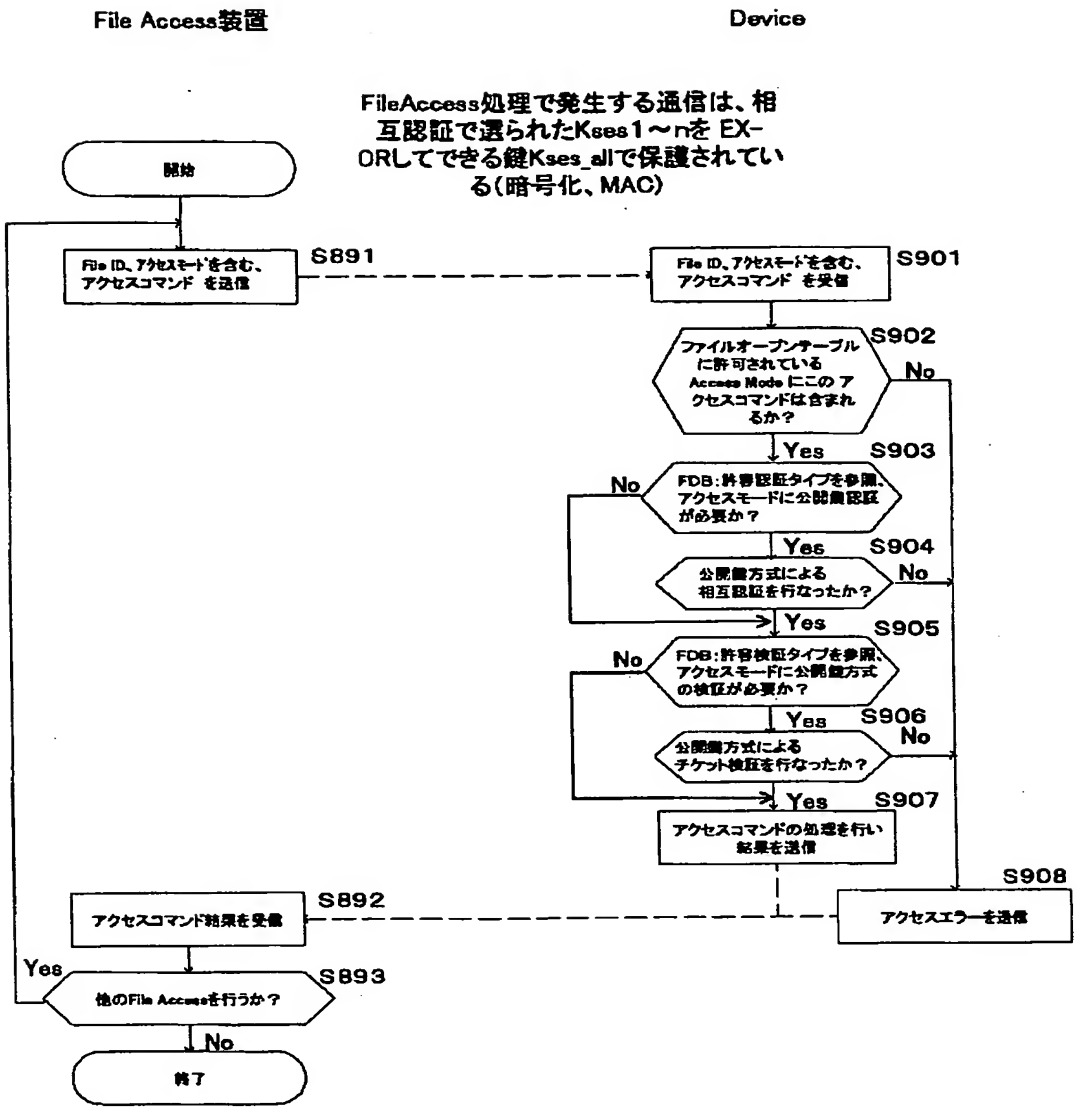
複合ファイル Structure Format

例） 電子マネー



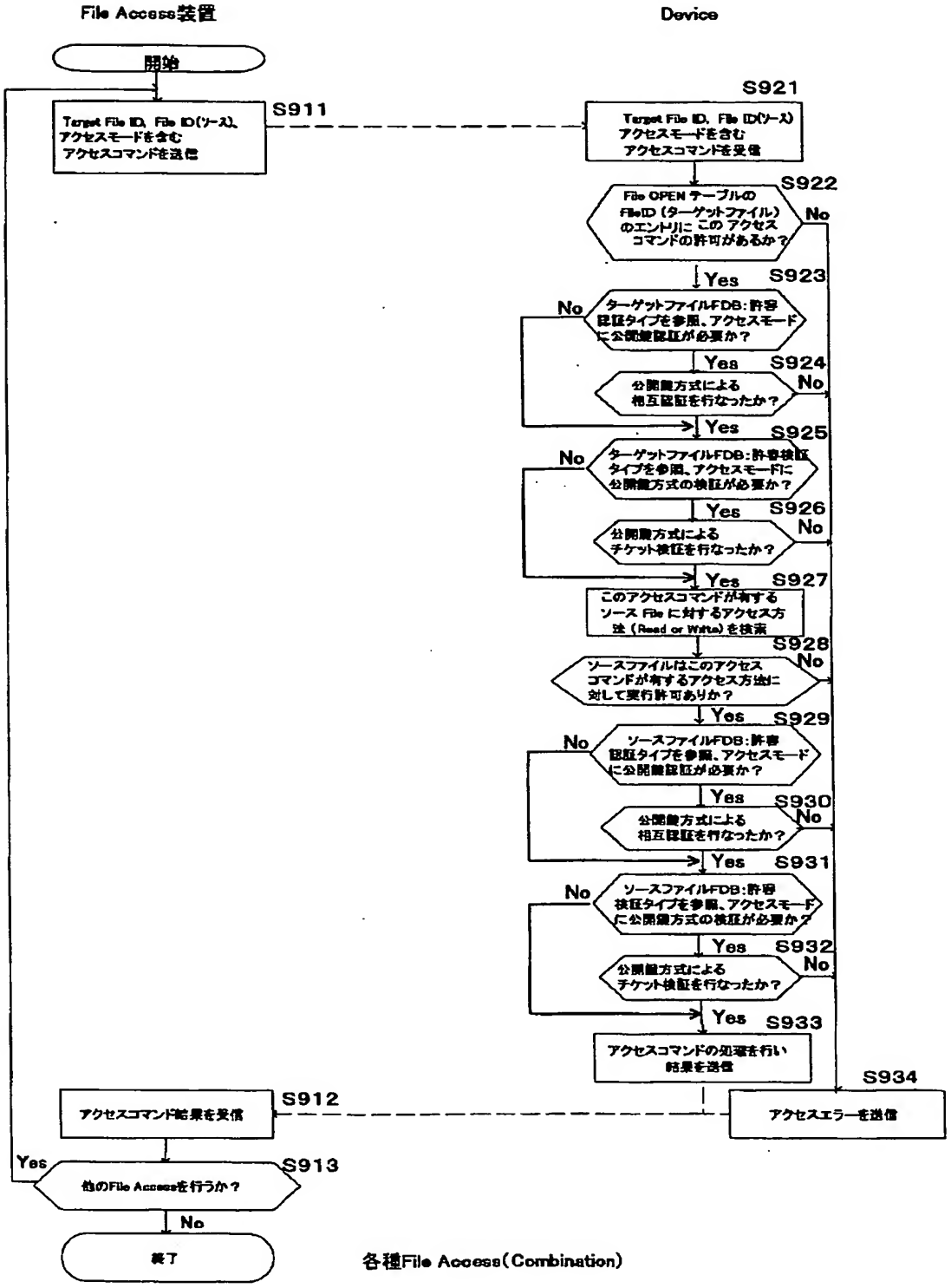


【図８６】

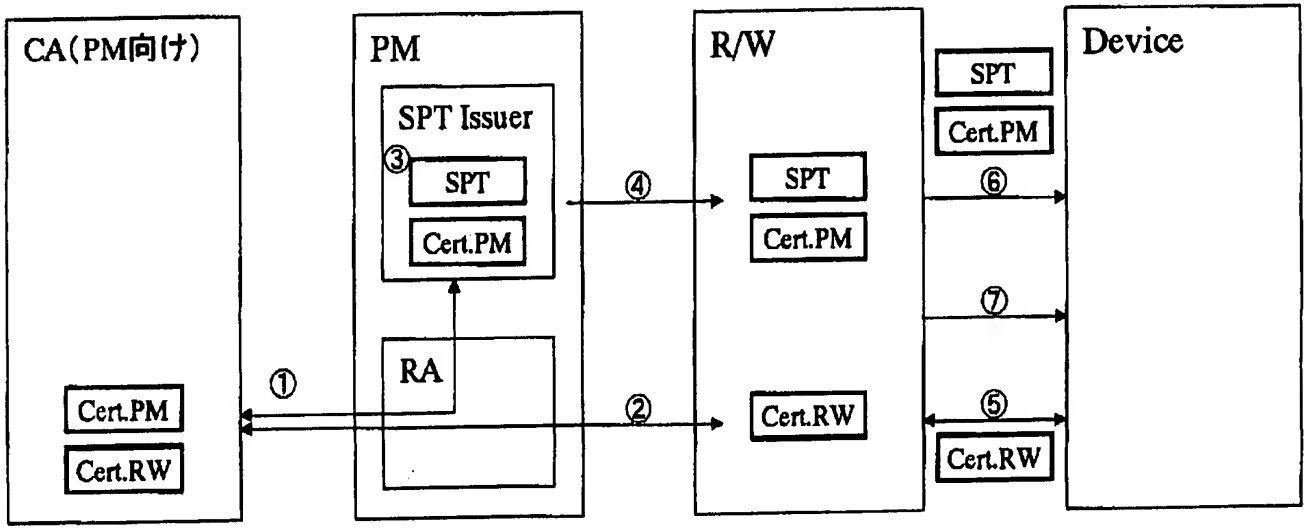


各種File Access(Normal)

【図８７】


File Access装置
Device
開始
S911
Target File ID、File ID(ソース)、
アクセスモードを含む
アクセスコマンドを送信
S921
Target File ID、File ID(ソース)
アクセスモードを含む
アクセスコマンドを受信
S922
File OPEN テーブルの
FileID (ターゲットファイル)
のエントリに この アクセス
コマンドの許可があるか？
No
Yes
S923
No
ターゲットファイルFDB:許容
認証タイプを参照、アクセスモード
に公開鍵認証が必要か？
Yes
S924
公開鍵方式による
相互認証を行なったか？
No
Yes
S925
No
ターゲットファイルFDB:許容検証
タイプを参照、アクセスモードに
公開鍵方式の検証が必要か？
Yes
S926
公開鍵方式による
チケット検証を行なったか？
No
Yes
S927
このアクセスコマンドが有する
ソース File に対するアクセス方
法 (Read or Write) を検索
S928
No
ソースファイルはこのアクセス
コマンドが有するアクセス方法に
対して実行許可ありか？
Yes
S929
No
ソースファイルFDB:許容
認証タイプを参照、アクセスモード
に公開鍵認証が必要か？
Yes
S930
公開鍵方式による
相互認証を行なったか？
No
Yes
S931
No
ソースファイルFDB:許容
検証タイプを参照、アクセスモード
に公開鍵方式の検証が必要か？
Yes
S932
公開鍵方式による
チケット検証を行なったか？
No
Yes
S933
アクセスコマンドの処理を行い
結果を送信
S934
アクセスエラーを送信
S912
アクセスコマンド結果を受信
Yes
S913
他のFile Accessを行うか？
No
終了
各種File Access(Combination)

Fileアクセス手順(Authentication Type: 公開鍵, SPT:公開鍵)



【図89】

①PM用の公開鍵証明書の発行
②R/W用の公開鍵証明書の発行
③SPT(Service Permission Ticket)の生成
④SPT及びPMのCertificateの供給
→SPTには、検証値(公開鍵)が付いている
⑤R/WとDeviceの間の相互認証(公開鍵)
⑥SPT及びPMのCertificateの送信
→SPTの検証
→SPT生成者の検証、SPT使用者の検証
→SPTに示すアクセス制限が解除される
⑦Fileへのアクセス

SPT例

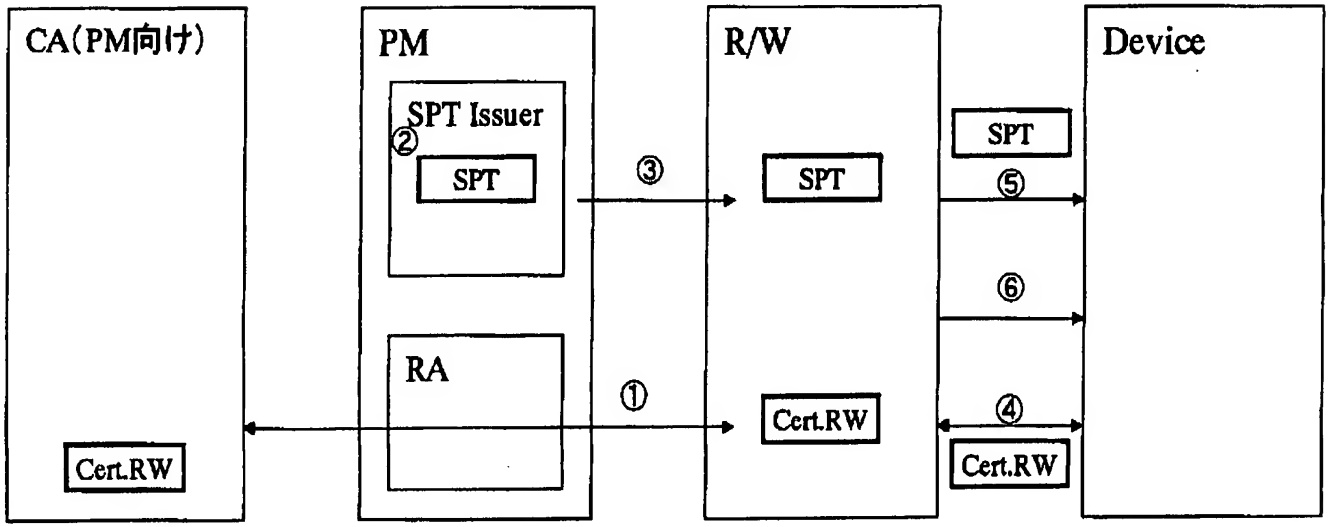
...
Authentication Type = 公
Category:SPT User
PMC1
File ID
File Access Mode
Signature

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC Category:SPT Issuer
Signature

特開平14-278838

Fileアクセス手順(Authentication Type: 公開鍵, SPT:共通鍵)



【図90】

①R/W用の公開鍵証明書の発行
②SPT(Service Permission Ticket)の生成
③SPTの供給
　→SPTには、検証値(共通鍵)が付いている
④R/WとDeviceの間の相互認証(公開鍵)
⑤SPTの送信
　→SPTの検証
　→SPT生成者の検証、SPT使用者の検証
　→SPTに示すアクセス制限が解除される
⑥Fileへのアクセス

SPT例

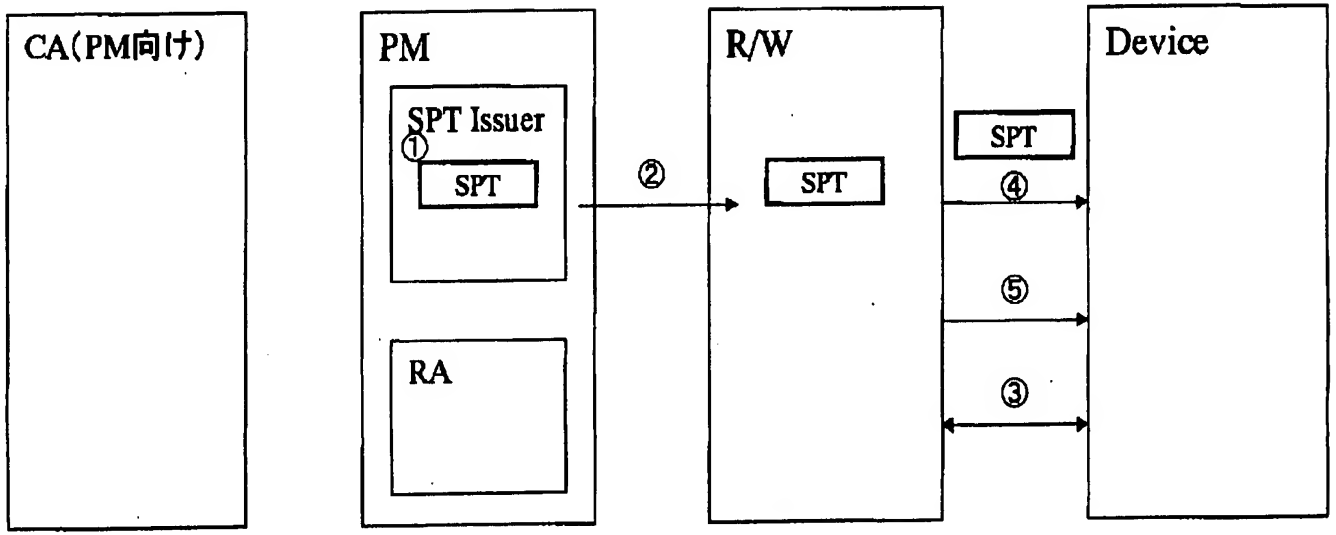
...
Authentication Type = 公
Category:SPT User
PMC1
File ID
File Access Mode
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC Category:SPT Issuer
Signature

特開平14-278838

Fileアクセス手順(Authentication Type: 共通鍵, SPT:共通鍵)



【図９１】

①SPT(Service Permission Ticket)の生成
②SPTの供給
　→SPTには、検証値(共通鍵)が付いている
③R/WとDeviceの間の相互認証(共通鍵)
④SPTの送信
　→SPTの検証
　→SPT生成者の検証、SPT使用者の検証
　→SPTに示すアクセス制限が解除される
⑤Fileへのアクセス

SPT例

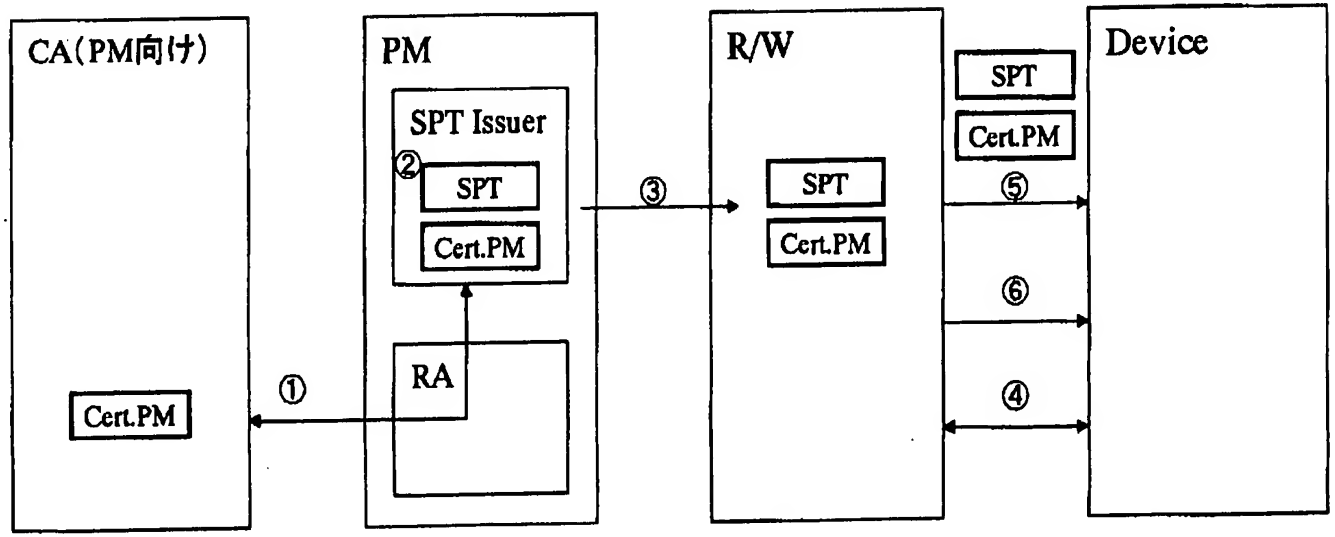
...
Authentication Type = 共
Category:SPT User
PMC1
File ID
File Access Mode
MAC

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC Category:SPT Issuer
Signature

特開平14-278838

Fileアクセス手順(Authentication Type: 共通鍵, SPT:公開鍵)



【図９２】

①PM用の公開鍵証明書の発行
②SPT(Service Permission Ticket)の生成
③SPT及びPMのCertificateの供給
→SPTには、検証値(公開鍵)が付いている
④R/WとDeviceの間の相互認証(共通鍵)
⑤SPT及びPMのCertificateの送信
→SPTの検証
→SPT生成者の検証、SPT使用者の検証
→SPTに示すアクセス制限が解除される
⑥Fileへのアクセス

SPT例

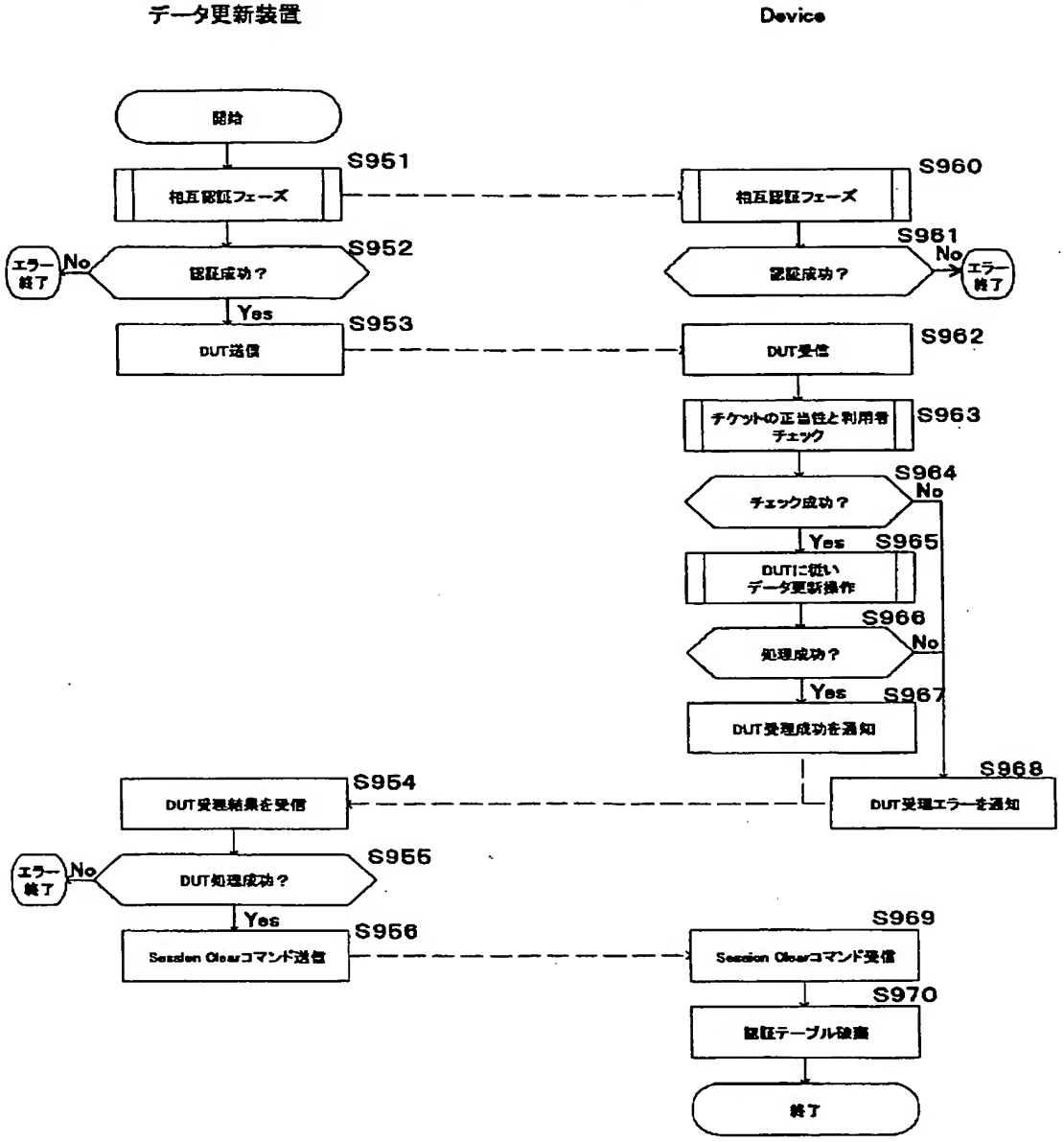
...
Authentication Type = 共
Category:SPT User
PMC1
File ID
File Access Mode
Signature

Certificate例

Certificate Version
SN
DN:Partition Manager
...
PM1 Public Key
Group:PMC Category:SPT Issuer
Signature

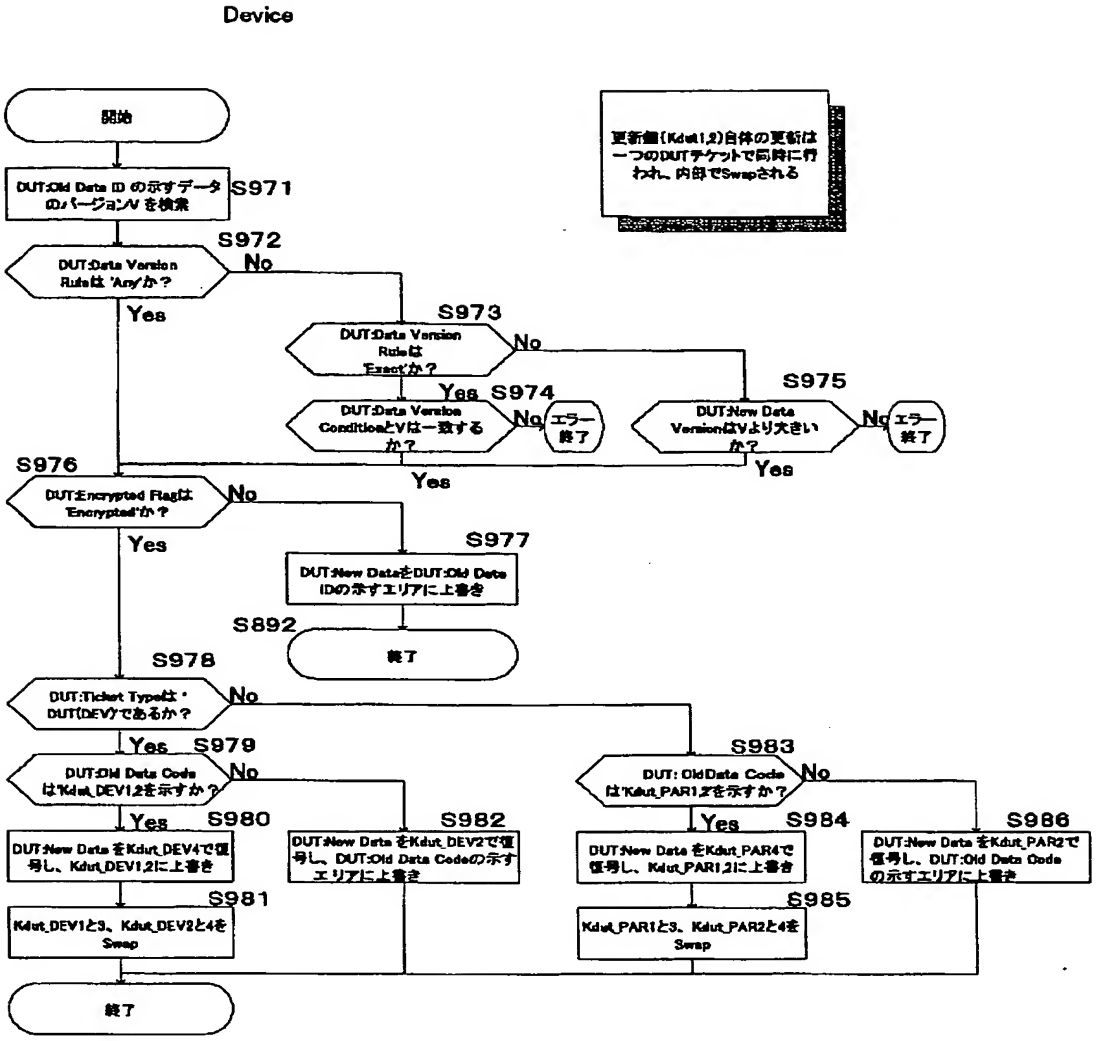
特開平14-278838

【図９３】


データ更新装置
Device
開始
S951
相互認証フェーズ
S960
相互認証フェーズ
S952
認証成功？
No
エラー終了
S961
認証成功？
No
エラー終了
Yes
S953
DUT送信
S962
DUT受信
S963
チケットの正当性と利用者チェック
S964
チェック成功？
No
Yes
S965
DUTに従いデータ更新操作
S966
処理成功？
No
Yes
S967
DUT受理成功を通知
S968
DUT受理エラーを通知
S954
DUT受理結果を受信
S955
DUT処理成功？
No
エラー終了
Yes
S956
Session Clearコマンド送信
S969
Session Clearコマンド受信
S970
認証テーブル破棄
終了

データ更新

【図９４】


Device
開始
DUT:Old Data ID の示すデータのバージョンV を検索
S971
S972
DUT:Data Version Ruleは 'Any'か？
No
Yes
S973
DUT:Data Version Ruleは 'Exact'か？
No
Yes
S974
DUT:Data Version ConditionとVは一致するか？
No
エラー終了
S975
DUT:New Data VersionはVより大きいか？
No
エラー終了
Yes
S976
DUT:Encrypted Flagは 'Encrypted'か？
No
Yes
S977
DUT:New DataをDUT:Old Data IDの示すエリアに上書き
S892
終了
S978
DUT:Ticket Typeは 'DUT(DEV)'であるか？
No
Yes
S979
DUT:Old Data Codeは'Kdut_DEV1,2'を示すか？
No
Yes
S980
DUT:New Data をKdut_DEV4で復号し、Kdut_DEV1,2に上書き
S981
Kdut_DEV1と3、Kdut_DEV2と4をSwap
S982
DUT:New Data をKdut_DEV2で復号し、DUT:Old Data Codeの示すエリアに上書き
S983
DUT: OldData Codeは'Kdut_PAR1,2'を示すか？
No
Yes
S984
DUT:New Data をKdut_PAR4で復号し、Kdut_PAR1,2に上書き
S985
Kdut_PAR1と3、Kdut_PAR2と4をSwap
S986
DUT:New Data をKdut_PAR2で復号し、DUT:Old Data Codeの示すエリアに上書き
終了
更新鍵(Kdut1,2)自体の更新は一つのDUTチケットで同時に行われ、内部でSwapされる

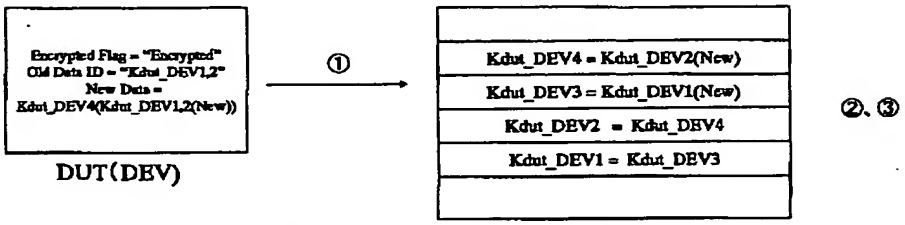
データ更新操作

【図９５】

Data Update

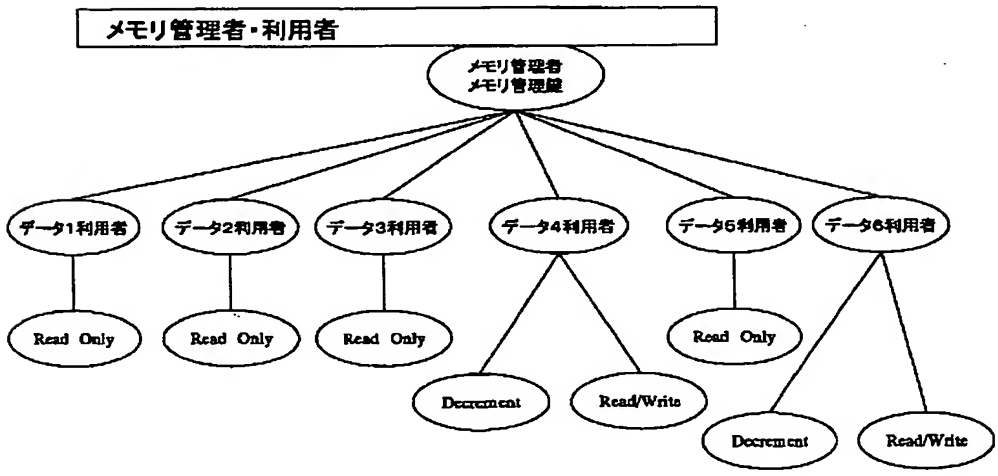
●更新対象が、Kdut_DEV1,2 や Kdut_PAR1,2 の場合 （Kdut_DEV1,2 として説明）

① 新しい Kdut_DEV1,2 (Kdut_DEV1(New), Kdut_DEV2(New))を Kdut_DEV4で暗号化。
Kdut_DEV4(Kdut_DEV1(New)), Kdut_DEV4(Kdut_DEV2(New))
DUTに記述、Deviceへ送信。
② Device は、Kdut_DEV1(New), Kdut_DEV2(New)を取り出し、古い Kdut_DEV1, Kdut_DEV2 を上書き
③ Kdut_DEV1, 3 と Kdut_DEV2,4 を入れ替え

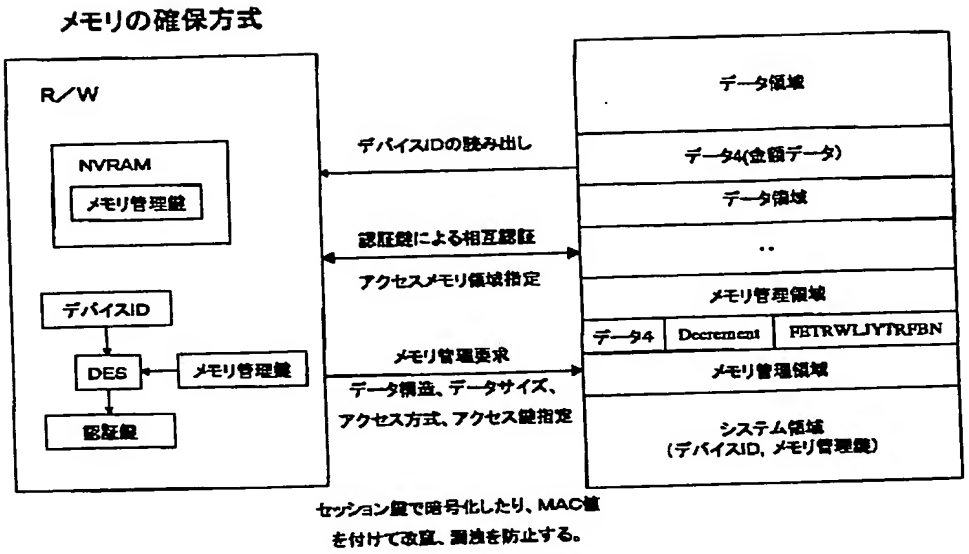


※ Kdut_DEV1,3 は、MAC用鍵、Kdut_DEV2,4 は、暗号用鍵
(Kdut_DEV1, KdutDEV2), (Kdut_DEV3, KdutDEV4) のペアで使われる。

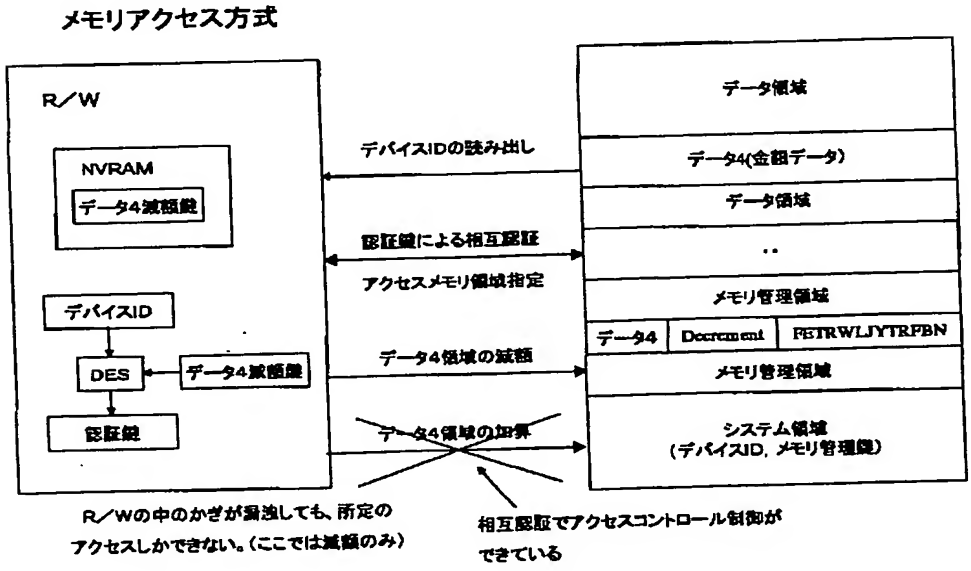
【図９７】



【図９８】



【図９９】



フロントページの続き

(51)Int.Cl.[7]		識別記号	FI		テーマコード(参考)
G06K	19/10		G09C	1/00	640Z　5J104
G09C	1/00	640			660C
		660	G06K	19/00	R
H04L	9/32		H04L	9/00	675D

(72)発明者　白井　太三
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　高田　昌幸
東京都品川区北品川６丁目７番35号　ソニー株式会社内

Ｆターム(参考)　5B017 AA07 BA06 BA07 BB09 CA11 CA14
5B035 AA13 BB09 CA11 CA38
5B058 CA27 KA02 KA04 KA08 KA31 KA35 YA20
5B082 EA11 GA11
5B085 AA01 AA08 AE01
5J104 AA07 AA16 EA05 KA02 MA03 NA02 NA05 NA35 NA38 NA42 PA12